Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1991-09

Design and implementation of a Nuclear Weapons Management System submodule : Shipboard Security Force System

Settlemyer, Sidney Russell

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/26453

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESIGN AND IMPLEMENTATION OF A NUCLEAR WEAPONS MANAGEMENT SYSTEM SUBMODULE: SHIPBOARD SECURITY FORCE SYSTEM

by

Sidney Russell Settlemyer

September, 1991

Thesis Advisor: Tung Bui

Approved for public release; distribution is unlimited

T258594

nclassified

ECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

a REPORT SECURITY CLASSIFICATION nclassified	1b RESTRICTIVE MARKINGS
a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
b DECLASSIFICATION/DOWNGRADING SCHEDULE	
PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL *(If applicable)* 55	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000		7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS *(City, State, and ZIP Code)*		10. SOURCE OF FUNDING NUMBERS

Program Element No	Project No	Task No.	Work Unit Accession Number

11. TITLE *(Include Security Classification)*

Design and Implementation of A Nuclear Weapons Management System Submodule: Shipboard Security Force System (u)

12. PERSONAL AUTHOR(S) Settlemyer, Sidney, R.

13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED From To	14 DATE OF REPORT *(year, month, day)* September, 1991	15. PAGE COUNT 136

16. SUPPLEMENTARY NOTATION

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18 SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*

DBMS application, ES application, Computer based PRP

19 ABSTRACT *(continue on reverse if necessary and identify by block number)*

The Nuclear Weapons Management System combines the strengths of an expert system with the flexibility of a database management system to assist the Weapons Officer, Security Officer, and the Personnel Reliability Program Officer in the performance of administrative duties associated with the nuclear wepaons programs in the United States Navy. This thesis examines the need for, and ultimately the design of, a system that will assist the Security Officer in administrative duties associated with the Shipboard Self Defense Force. This system, designed and coded utilizing dBASE IV, can be implemented as a stand alone system. Furthermore, it interfaces with the expert system submodule that handles the PRP screening process.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Tung Bui	22b. TELEPHONE *(Include Area code)* 408-646-2630	22c. OFFICE SYMBOL AS/Bd

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Approved for public release; distribution is unlimited.

Design and Implementation of a Nuclear Weapons Management System Submodule: Shipboard Security Force System

by

Sidney R. Settlemyer
Lieutenant, United States Navy
B.S.,Savannah State College, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

ABSTRACT

The Nuclear Weapons Management System combines the strengths of an expert system with the flexibility of a database management system to assist the Weapons Officer, Security Officer, and the Personnel Reliability Program Officer in the performance of administrative duties associated with the nuclear weapons programs in the United States Navy. This thesis examines the need for, and ultimately the design of, a system that will assist the Security Officer in administrative duties associated with the Shipboard Self Defense Force. This system, designed and coded utilizing dBASE IV, can be implemented as a stand alone system. Furthermore, it interfaces with the expert system submodule that handles the PRP screening process.

5418625

TABLE OF CONTENTS

I. INTRODUCTION 1

A. BACKGROUND 1

B. RESEARCH QUESTIONS 1

C. OBJECTIVES 2

D. SCOPE, LIMITATIONS, and ASSUMPTIONS 3

E. SUMMARY OF FINDINGS 4

II. GENERAL DESCRIPTION OF THE NUCLEAR WEAPONS MANAGEMENT SYSTEM 6

A. BRIEF OVERVIEW OF THE PERSONNEL RELIABILITY PROGRAM 6

B. BRIEF OVERVIEW OF THE SHIP'S SELF DEFENSE FORCE 8

C. COMPUTER AIDS SUITED FOR MANAGEMENT OF PRP AND SSDF 11

III. DESCRIPTION OF THE NWMS / SSFS SUBSYSTEM 14

A. COMBINED SYSTEM OVERVIEW 14

B. DESCRIPTION OF THE SECURITY FORCE SYSTEM . . . 19

1. Requirements Option Menu 20

a. View Specific Requirement 21

b. Edit Requirement 22

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

c. Print Requirement 23
2. Roster Menu 23
a. View Security Roster 24
b. Update Security Roster 25
c. Print Security Roster 25
3. Drills Menu 25
a. View Drills 26
b. Edit Drills 27
c. Print Drills 28
4. Print Security Force 28
a. Print Requirements 29
b. Print Roster 30
c. Print Drill Report 30
d. Print Qualifications Nearing Expiration 31
e. Print Expired Results 32
5. Qualifications 34
a. List Qualifications Near Expiration . . 35
b. Expired Qualifications 36
6. Utilities 36
a. Change System Date 37
b. Backup Drive A 38
c. Download Application 39
d. Repair Indexes 39
7. Exit 39
a. Exit to dBASE IV Control Center 40
b. Exit to Dos 41

IV. IMPLEMENTATION CONSIDERATIONS 42

A. SHIP'S SECURITY FORCE SYSTEM AS AN NWMS SUBMODULE 42

B. SHIP'S SECURITY FORCE SYSTEM AS A STAND ALONE APPLICATION 43

C. FUTURE EXPANSION POSSIBILITIES 43

V. CONCLUSIONS AND RECOMMENDATIONS 46

APPENDIX A: REFERENCES 48

APPENDIX B: USER MANUAL. 52

APPENDIX C: DATA DICTIONARY. 53

APPENDIX D: PROGRAM CODE 55

INITIAL DISTRIBUTION LIST 128

I. INTRODUCTION

A. BACKGROUND

The purpose of this thesis is to examine the programs, identify the requirements, and create an automated system to assist fleet personnel in the execution of administrative duties mandated by the Nuclear Weapons Management Programs and Shipboard Security in the U.S. Navy. The major focus of this thesis is to develop a PC-based application to automate an otherwise time consuming, manpower intensive, laborious process required aboard every commissioned United States Naval Warship.

B. RESEARCH QUESTIONS

The following questions will be answered in the research, design and ultimate development of the thesis:

1. What areas of nuclear weapons administration are appropriate for computer systems to aid in decision making, as well as information evaluation and tracking?

2. Once specific areas are identified, what types of computer systems and software application packages are suitable, readily available, and most importantly appropriate for the needs of this particular problem?

3. What is the ideal design of the software package that will readily support the organization of all training, qualification, and administrative requirements?

4. How will the database management system and the expert system modules integrate to support the Nuclear Weapons Management System?

C. OBJECTIVES

The objectives of this thesis are as follows:

1. Develop a database system that will enable fleet personnel to optimize resources in tracking personnel security data, and perform such administrative tasks as file manipulation, report generation, and qualification update and review.

2. Design the database system such that it can be used not only as a sub-module for the Nuclear Weapons Management System [Ref. 1], but in a stand alone, non-nuclear application as well.

3. Provide discussions on what the Personnel Reliability Program (PRP) and the Ship's Self Defense Force (SSDF) are, and how software packages can aid the fleet in the many administrative tasks that are required to ensure safe, and secure accomplishment of the Navy's mission.

4. Provide a discussion on the interoperability of the Nuclear Weapons Management System [Ref. 1], and the Ship's Security Force System. The design and implementation

portions of this thesis will use two software packages (dBASE IV and VP-Expert) and the integration of the two packages to support a combined system.

D. SCOPE, LIMITATIONS, and ASSUMPTIONS

There are four areas of the NWMS that are of major concern to any command involved with Nuclear Weapons Management Programs in the Navy. These areas include:

- Personnel Reliability
- Technical Operations
- Training
- Security

Personnel Reliability is the thrust of another thesis being conducted (NUCLEAR WEAPONS MANAGEMENT SYSTEM Ref. 1); This thesis will be focusing on the security issue.

Moreover, this thesis will concentrate mainly on a database system developed to aid fleet users in the manipulation of information in conjunction with the Nuclear Weapons Management System [Ref. 1], and the Security Force programs. The system is designed to operate on IBM compatible microcomputers that are prevalent in the fleet today. To operate the Security Force System, dBASE IV will be required to be in place on the shipboard microcomputer; to operate the combination of the Nuclear Weapons Management System (NWMS) and the Ship's Security Force System (SSFS), both dBASE IV and VP-Expert must be in place on the microcomputer. The above software will operate on an 8088 XT microcomputer; however it is recommended that the system be run on at least a 80286 AT system because of enhanced speed and computing power. The 80286-based Zenith 248 computers currently installed on ships in the fleet are adequate for the proposed application.

E. SUMMARY OF FINDINGS

As a result of the research conducted, published inspection results, and informal interviews with a variety of fleet personnel, the following issues were found to be relevant:

1. There exists a real need, fleet wide, for a system to assist in the administrative duties involved in Nuclear Safety and Shipboard Security programs that accompany not only Nuclear capable ships, but also non-nuclear capable ships.

2. dBASE III+ and dBASE IV are readily available in the fleet today, they are compatible with the software utilized in the Nuclear Weapons Management System [Ref. 1], and are flexible enough to allow adaptation to a variety of command applications. Consequently, they are a good choice in software packages to assist in fulfilling general administrative requirements.

3. An automated system consisting of dBASE IV and VP-Expert (Chapter II identifies the need for 2 software

packages) can be designed and built to assist the fleet in management of Nuclear Weapons Safety and Security programs, and as such can be implemented on existing fleet hardware.

4. The automated subsystem consisting of dBASE IV software can be developed in such a way as to be functional in a stand alone, non-nuclear application for SSDF, and can also be implemented on existing fleet hardware.

II. GENERAL DESCRIPTION OF THE NUCLEAR WEAPONS MANAGEMENT SYSTEM

A. BRIEF OVERVIEW OF THE PERSONNEL RELIABILITY PROGRAM

Nuclear weapons management is by far one of the most serious and important tasks in the U.S. Navy today. Safety concerns and public awareness demand that this system be accurate, current, and reliable at all times. The responsibility for these programs resides with every afloat command in the United States Navy, and virtually all shore commands. It is for these reasons that strong management techniques must be applied, and rigid inspections be performed to monitor compliance.

The Chief of Naval Operations, through OPNAV Instructions, publishes guidelines for the Navy to utilize for the programs discussed. Weapons Training Groups on both East and West Coasts are tasked with training the fleet to ensure compliance with current directives as well as conducting inspections on Nuclear capable ships and stations. Sanitized inspection results are published periodically as WPNTRAGRU notices. Close scrutiny of inspection results clearly demonstrates the need for administrative improvements which can be easily effected through computerization of all routine administrative tasks. As with the PRP screening function, off the shelf software packages and existing hardware can easily facilitate the rapid prototyping of a system that can easily automate all standardized administrative functions.

The Personnel Reliability Program (PRP) was established to ensure that personnel who work with nuclear weapons, either directly or indirectly, are trustworthy and reliable. This of course is not a process that can be established with a single entry level screen. The importance of the nuclear weapons program, and the potential for accidental or willful discharge of nuclear weapons, demands that the screening of individuals be accomplished on a recurring basis. This process involves not only entry level, in-depth investigations and periodic reviews of personnel records, but a routine systematic review and analysis of personnel already in the PRP system.

During the process of initial screening, there are many questions that must be answered in determining a candidate's qualifications for the PRP. The many iterations of questions and answers lend themselves well to the use of an expert system. This system as indicated earlier is the basis of another thesis [Ref. 1] in which the Ship's Security Force System can be utilized as a sub-module or subsystem. The myriad of requirements that the PRP screening process includes are: security clearances for the position held, physical qualifications, mental state, legal and military law infractions, and personal beliefs. These items are covered much more in depth in Ref. 1; however, since this thesis covers only PRP in relation to a database subsystem, further discussion of the use of expert system technology can be found in [Ref. 1]. Database systems are currently utilized in the fleet by many department heads and division officers to track administrative items on department and division personnel. It is therefore not only a suitable alternative to use, but a proven solution as well. The fact that dBASE IV is readily available in the fleet clearly makes it the system of choice for this application because the training curve for utilization should not be as high as other less familiar software packages would be.

B. BRIEF OVERVIEW OF THE SHIP'S SELF DEFENSE FORCE

All ships in the fleet today, whether nuclear or non-nuclear capable, have a Self Defense Force. It is with this in mind that the system designed for this thesis be able to function as both a subsystem and a stand alone system. The Ship's Self Defense Force (SSDF) is designed to defend the ship from a hostile boarding or intrusion. The force consists of four elements, the Security Alert Team (SAT), the Backup Alert Force (BAF), the Reserve Force (RF), and the Augmentation Force (AF).

The Security Alert Team is defined as two or more security force members, in addition to sentries at established posts, capable of responding immediately to any physical security irregularity [Ref. 2]. Current directives require the Security Alert Team to be capable of responding within five minutes from the sound of the alarm.

The Backup Alert Force is defined as three or more designated personnel; armed, equipped, trained, and capable of providing assistance to the Security Alert Team within ten minutes [Ref. 2]. The main purpose of the BAF is to support the SAT.

The Reserve Force is defined as ten or more designated personnel; armed, equipped, trained and capable of responding in support of the on-duty security force personnel within fifteen minutes [Ref. 2]. The main purpose of the Reserve Force is to support both the SAT and the BAF.

The Augmentation Force is defined as a group of personnel organized and designated in established plans by fleet and area commanders to best facilitate and complement individual type ship requirements [Ref. 2]. Generally, the Augmentation Force is comprised of the remainder of the ship's crew that is aboard at the time. This further stresses the importance of the requirement that all hands be familiarized in the use of firearms and security procedures.

The definitions alone highlight several of the items that must be recorded and tracked to ensure compliance with directives as well as to ensure safety. Each member of the security force is required to be qualified and trained on the weapon or weapons that he or she may have to use during a security alert. In particular, each of the security force members must be qualified on four weapons: the .45 caliber pistol, the 9mm pistol, the M14 or M16 semi automatic rifle, and the 12 gauge pump shotgun. These qualifications are required to be updated once per year, with the procedures ranging from training lectures to actual live target fire at shore based ranges. Qualifications and training for the security force as a whole exist in the form of required schools and drill scenarios. Each member of the ship's security force is required to attend a shore based school on shipboard security tactics. Furthermore, each duty section aboard ship is required by current directives to be drilled once per week. These drills consist of realistic scenarios that must be responded to and trained for in order to ensure maximum readiness and peak proficiency at all times. Close attention must paid in order to ensure that each member has been given proper training to safely handle the appropriate weapons. Requirements for annual qualifications on small arms weapons, training, military instruction on weapons handling, use of deadly force, and other security tactics are all examples of security management items that must be constantly reviewed and updated to ensure not only compliance with regulations, but safety and reliability as well. Documenting the satisfactory completion of these requirements is related more to book keeping and time line functions; therefore it is well suited to computer application. Another important aspect is that these directives do not specify which divisions and/or departments will provide the requisite people. Consequently, the Security Force is often made up of personnel from all departments and divisions. This complicates the administrative burden with regard to planning for replacement personnel when security force members rotate off the ship, and arranging qualification certification. These functions are nearly impossible without some type of automated assistance.

On ships that are nuclear capable, these problems are exacerbated by the requirement for personnel who perform security duties in proximity of nuclear weapons to be PRP qualified. The SAT and BAF personnel must be screened into the PRP, since they are the first and second teams to respond, and may have access to nuclear weapons on ships. The other security force members, although not required to be PRP screened, must be continually monitored for the same qualifications and training required of PRP security force personnel.

C. COMPUTER AIDS SUITED FOR MANAGEMENT OF PRP AND SSDF

From the research that was conducted on PRP screening [Ref 1] it became readily apparent that an expert system was the best option available. The design of the PRP screening process is essentially a series of well-defined questions and answers. There are a variety of requirements that a person must meet to be screened into the PRP. Some of these requirements are concerned with physical condition, substance abuse, military and civil law infractions and overall mental state. These requirements are complex. Depending upon the answers to specific questions, they can lead to further questioning for clarification and/or different degrees of qualification. This rule based qualification procedure corresponds directly with optimal utilization of an expert system. Used heavily in the medical and engineering fields, an expert system is designed to ask the user a set of questions and based upon the user's answers to those questions and the rules that are stored within the expert system, the expert system will provide the user with answers. These answers can be constructed in a simple "yes/no" format to something more elaborate as providing lengthy answers with referenced proof materials. This application is suitable for the PRP screening process because it will ask preselected questions of the person doing the screening process, and will provide advice on the candidate's qualifications for the PRP. It also will reference current directives and instructions when a negative response for entry into PRP is warranted.

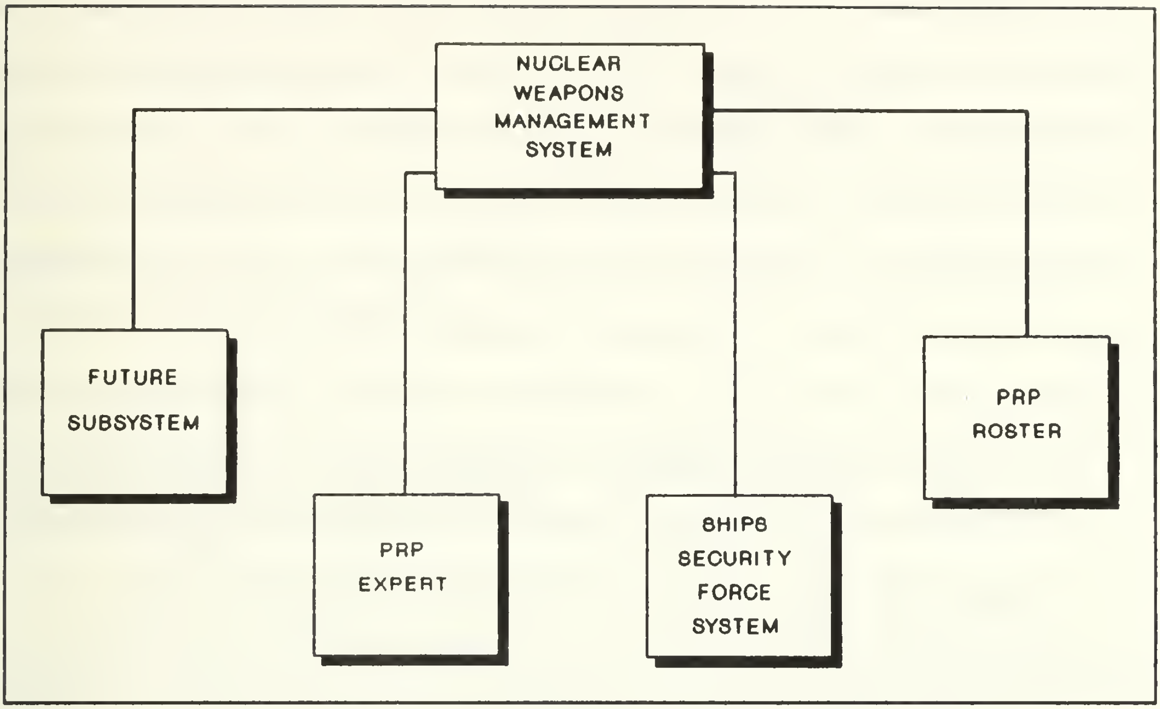
Once the screening process is completed, the need still exists to store and, perhaps at a later date, manipulate the data to meet the needs of the user. Although VP-Expert, as an inferencing engine, is a very good application for the screening process, it does not offer database manipulation and storage facilities. It does however interact well with dBASE IV, which is another software package that is designed primarily for this purpose. A database system, such as dBASE IV, is designed to accomplish many types of file manipulation, form generation, report generation, as well as querying information and even generation of labels and code. These are the factors important to the user when tracking this information, whether used in conjunction with PRP or not.

III. DESCRIPTION OF THE NWMS / SSFS SUBSYSTEM

A. COMBINED SYSTEM OVERVIEW

In this chapter, the Nuclear Weapons Management System is discussed. It is important to realize that this system, designed in conjunction with two theses, contains several parts. Three of these parts are currently completed in design, and the remainder of the parts or sub-modules, will be discussed in the next chapter of this thesis under the heading of future expansion possibilities.

The Nuclear Weapons Management System, in its current state, consists of a database, an expert system, and two database applications. The expert system module employed is utilized to screen potential PRP candidates. Once this screening process is completed, an option to enter a successful candidate into the PRP database is given, and upon acceptance of the option the information on the candidate is then passed on and stored into the database. This information can be further manipulated through one of two applications in the form of reports, entry lists, qualification lists, general information, edit functions and techniques. A diagram of the Nuclear Weapons Management System and relational software packages is provided below. The Nuclear Weapons Management System:


NUCLEAR WEAPONS MANAGEMENT SYSTEM
FUTURE SUBSYSTEM
PRP EXPERT
SHIPS SECURITY FORCE SYSTEM
PRP ROSTER

Once the NWMS software has been loaded onto the microcomputer that has dBASE IV and VP-Expert installed, the paths properly set and execution file entered (these instructions are provided in the users manual appendix), the user will immediately be offered a menu to select the option of entering the PRP screening process, the PRP database, the Security Force database, or returning the computer to the DOS environment. The PRP screening option has been coded such that when selected, the user is automatically directed into the VP-Expert software environment. At this point, instructions are given to select the PRP screening function, and when completed, the user is presented with the first of many options and questions that exist in the expert system's knowledge base. The information programmed into the knowledge base of the expert system has been extracted from current directives as to the fitness and qualifying factors required for PRP candidacy. This information in conjunction with VP-Expert's rule based system of interpretation will then formulate the system's recommendations as to whether or not the information entered complies with current directives, and hence represents a viable candidate for the PRP. If the system returns an unfavorable decision based on the information reported, specific disqualifying references are presented to the user for either correction or justification.

The second option in the NWMS menu is the Ship's Security Force Database Application. Upon selection of this option, the user is mapped through dBASE IV, directly into the application. A sign on banner is presented that notifies the user that he or she is now in the Security Force application. To proceed, the enter button is depressed to reveal a main menu that provides a variety of options from reading fleet requirements on security, to updating and printing rosters, and drill reports. This application will be discussed in further detail in part B of this chapter of the thesis.

The third option in the NWMS menu is the PRP Database Application. This option is coded such that once again the user is mapped through the dBASE IV software and directly into the PRP Database Application. A sign on banner is again presented to notify the user that he or she is now in the PRP Database Application, and with a depression of the enter button the user is presented with a main menu that consists of four options. The options herein include View PRP Roster, Edit PRP Roster, Print PRP Roster, and Exit. A sample menu diagram is provided below.

VIEW ROSTER	EDIT ROSTER	PRINT ROSTER	EXIT

The first option, View PRP Roster, allows the user to quickly review all of the members of the PRP in alphabetical order. This option is very helpful to the PRP Officer because it allows quick access for checking any number of specific personnel in either controlled or uncontrolled billets within the PRP. It also assists the PRP Officer in areas such as continual update and rescreening of personnel. The layout or format of the screen presentation is tabular to provide quick, easy viewing. This option is accomplished in dBASE IV by programming a view option within the software to retrieve the information from the database file (mem_info.dbf).

The second option on the menu, Edit PRP Roster, allows the PRP Officer to not only access the information, but edit it as well. This is an important option because it allows errors that exist in record keeping to be corrected easily and without delay. With the amount of work that must be accomplished, and the limited time available, typographical errors seem to abound in the fleet. This option is also much better designed for a dBASE application, than for that of the expert system. Errors are much more prone to be discovered after the initial screening process is completed, when the data is being reviewed, and this option exists to correct them. This Edit option is accomplished within the software by allowing the user to append the database file (mem_info.dbf).

The third option on the menu, Print PRP Roster, allows the PRP Officer to maintain a hard copy report of the PRP. This is extremely important aboard ship with unexpected power losses, power fluctuations, and maintenance, the possibilities of losing important documentation at least temporarily are very real. This option is also beneficial to the PRP officer because it allows quick access to hard copy reports or data that may be required by the chain of command. It seems at times that in the shipboard environment, seeing is believing, and coming to the PRP office, or PRP officer's stateroom to view a report that is stored on computer is not a viable option. This menu option is accomplished in the software by designing a report format within dBASE IV, and then calling that report to be printed.

This report also retrieves it's data from the database file (mem_info.dbf).

The fourth and final option on the menu, Exit, simply returns the user back to the NWMS main menu. This is accomplished in the code by simply directing a path from the dBASE IV software package to DOS batch file that makes up the NWMS menu.

The final option in the NWMS menu is that of returning to the DOS function. This option is provided more for the casual computer user than anything else, because in reality, the batch file that makes up the NWMS main menu exists in DOS. It seems redundant to offer this option to the experienced user who will probably know that he or she is in DOS anyway; however, the design of the system must take into account the varying degree of computer literacy in the fleet.

B. DESCRIPTION OF THE SECURITY FORCE SYSTEM

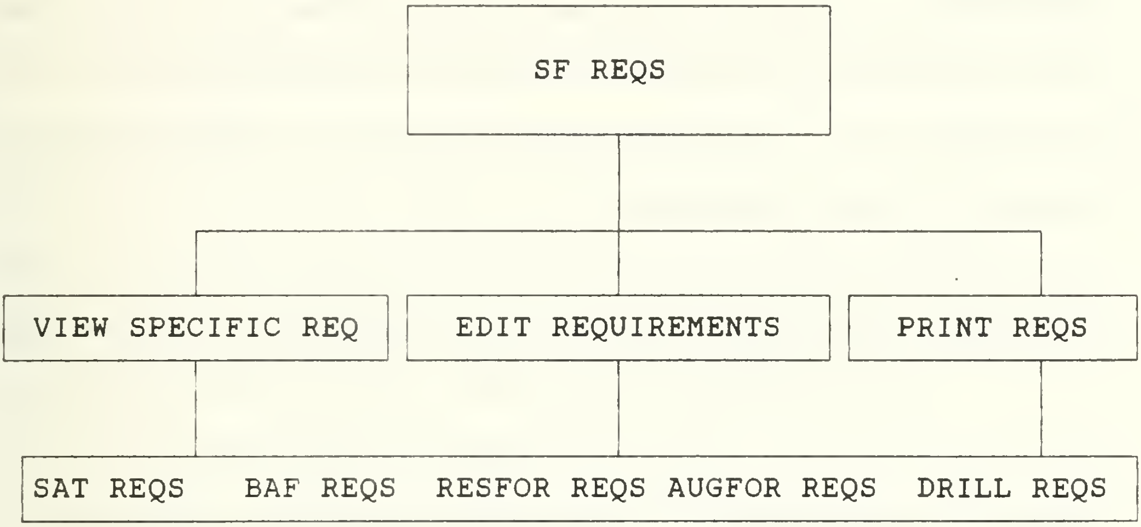
The Ship's Security Force System is a menu driven, dBASE IV application that provides the user with many data manipulation techniques and functions. It is menu driven so that as the user progresses, menus are provided until the user either reaches a level to access data or exits the program. This application can be used in conjunction with the NWMS as a subsystem, as previously described, or as a stand alone Security Force System. To use this application as a stand alone entity, it is important to realize that the PRP screening functions discussed earlier cannot be accomplished in this application. The file and data manipulation of Security Force, as well as PRP personnel can be accomplished and will be discussed in depth in this chapter.

In the user's manual, instructions on how to load the dBASE software security applications in a stand alone mode are provided. Once these instructions are implemented, the user will be presented with a sign on banner that assures them that they are in the Shipboard Security Force System. By depressing the enter key they are presented with the main horizontal bar menu (MEMMAIN) to begin their selection of options. A sample menu diagram is provided below.

SF REQS SF ROSTER SF DRILLS PRINT SF QUALS UTILS EXIT

1. Requirements Option Menu

The first option available to the user from the main menu is entitled SF REQS. This acronym is defined as Security Force Requirements. The requirements that are being discussed are those requirements obtained from current directives that pertain to security force matters and will be introduced in this chapter. A layered menu diagram is provided below.



When the SF REQS option is chosen, the user will be presented with an additional pop up menu of selections (REQSSF). This pop up menu of additional selections includes the abilities to either view, edit or print a requirement.

A sample menu diagram is provided below.

```
VIEW SPECIFIC REQUIREMENT ->
EDIT REQUIREMENTS         ->
PRINT REQUIREMENTS        ->
```

The three arrows included in the pop up menu above are used as pointers to let the user know that there are additional menu options layered below each option as shown.

a. View Specific Requirement

As in the case above, the first option, VIEW SPECIFIC REQUIREMENT, presents the user with a final horizontal bar menu (REQSALL) that allows the user to choose which requirement he or she would like to view. A sample menu diagram is provided below.

SAT REQS   BAF REQS   RESFOR REQS   AUGFOR REQS   DRILL REQS

The above options, when chosen will provide the user with specific requirements as set forth in OPNAV and Weapons Training Group instructions. The selections are acronyms for the following: SAT REQS - Security Alert Team requirements, BAF REQS - Backup Alert Force requirements, RESFOR REQS - Reserve Force requirements, AUGFOR REQS - Augmentation Force requirements, and DRILL REQS - those requirements that define periodicity of drilling the security force. These options are accomplished by selecting the internal software options to retrieve the information from the database file (reqs.dbf). These files had to constructed in the form of memo fields in order to manipulate text instead of data.

b. Edit Requirement

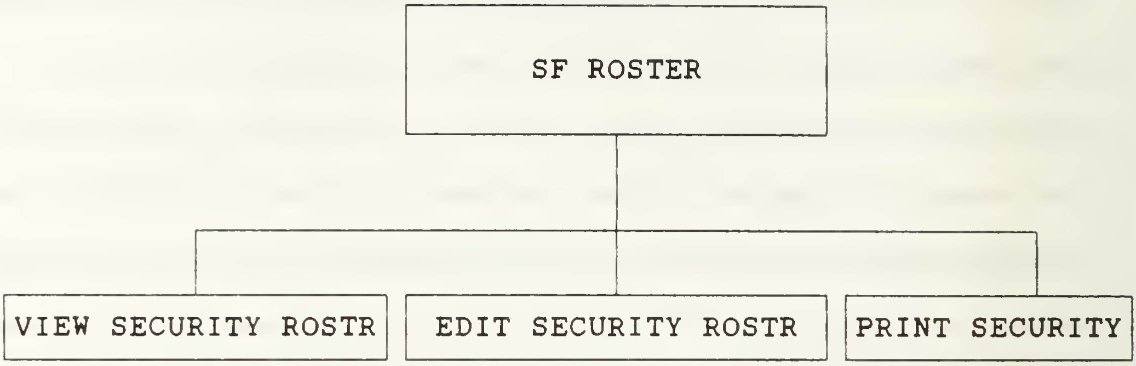
The second option on the REQSSF pop up menu, EDIT REQUIREMENTS, allows the user to update the individual requirements as the changes are received from the chain of command. When this menu is selected, the user is once again presented with a horizontal bar menu (EDREQS) that is identical to the horizontal bar menu (REQSALL) previously discussed. The major difference between the two options is that the user is allowed to make changes to the text in the option area selected. This is accomplished much in the same way as the view option in that it retrieves the text from the database file, but then differs in allowing changes to be made by calling up a previously designed form.

c. Print Requirement

The third and final option of the REQSSF pop up menu, PRINT REQS, allows the user to print the requirement of choice. Once again the user is presented with an additional horizontal bar menu (PRNTREQS) and can choose which specific requirement he or she would like to print. The PRNTREQS menu is identical to the two previous menus (REQSALL, EDREQS) and the structure of this function is very similar. The software retrieves the text from the database file and sends the print instruction to the printer.

2. Roster Menu

Returning our attention to the main menu of the Security Force System, we will discuss the next option, SF ROSTER. This option, an acronym for Security Force Roster, allows the user to manipulate data on personnel included in the security force. A layered menu diagram is provided below.



With its selection, the user is presented with another pop up menu (ROSTER), which will allow the user to view, update, or print the security force roster. A sample menu diagram is provided below.

VIEW SECURITY ROSTER
UPDATE SECURITY ROSTER
PRINT SECURITY ROSTER

a. View Security Roster

Selection of the first option will allow the user to view a list of all Security Force personnel in alphabetical order. This option will allow the Security Officer or Weapons Officer a much needed, quick access to pertinent data (e.g. how many people are currently on the security force, when did specific personnel report aboard, and more importantly when are they going to detach from the command). All of these items are convenience features that support optimal management of Security Force Administration. This option, VIEW SECURITY ROSTER, is accomplished by retrieving the data directly from the database file in which it is stored (mem_info.dbf).

b. Update Security Roster

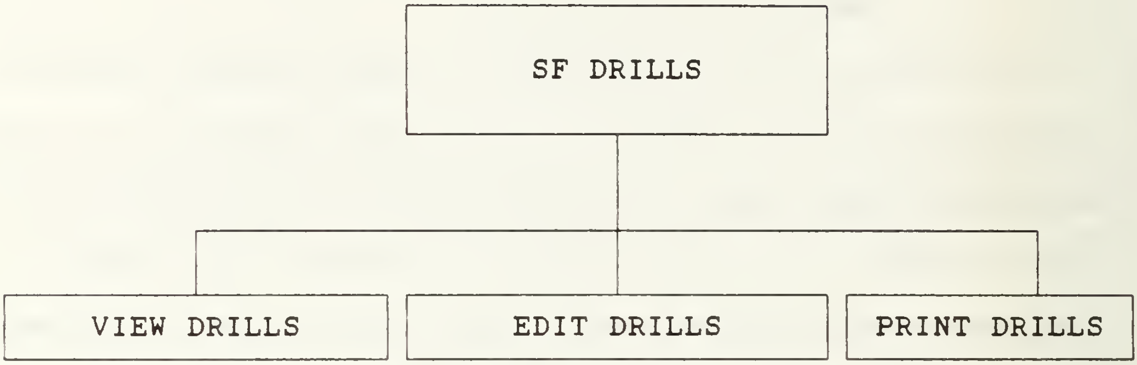
Selection of the second option, UPDATE SECURITY ROSTER, allows the user to append or correct the data that is stored in the dbf file. This is accomplished by retrieving the data from the dbf file, and allowing the changes to be made through a previously designed form in the dBASE IV software. The benefit of this option is that the Security Officer does not need to reenter all the data on an individual, he or she can make only those changes necessary.

c. Print Security Roster

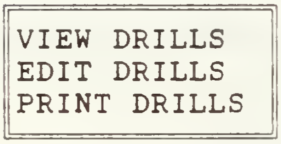
Selection of the third and final option for this pop up menu, PRINT SECURITY ROSTER, will allow the user to print a list of all members of the Security Force. This is a valuable tool with respect to requirements for hard copy data in the fleet. Accomplishment of this option is carried out by mapping the internal dBASE software through a previously prepared report format, and sending the print instructions to the printer.

3. Drills Menu

Once again returning to our main menu (MEMMAIN) we proceed to the third option, SF DRILLS. This option allows the user to view, edit or print Security Force drills that current directives require to be accomplished and recorded. A layered menu diagram is provided below.



Upon selection of this option, the user is presented with another pop up menu (DRILLS). A sample menu diagram is provided below.



a. *View Drills*

Selection of the first of these options, VIEW DRILLS, will allow the user to view the historical data of security force drills that have been entered into the database. This option presents a valuable tool for the Security Officer in which to study trends in drilling techniques, along with redundancy of specific scenarios in drilling. It is not wise to drill the security force on solely one or two drill scenarios or even in one or two areas of the ship. With this feature implemented in the Security Force System, redundancy of training can be easily avoided. This feature also allows the Security Officer to review historical security alert drill reports and to study actual security alerts. These actual alerts are also entered in this system and are commonly utilized in the fleet as training tools. This option is accomplished within the system software by mapping this menu selection to the database file (drills.dbf).

b. Edit Drills

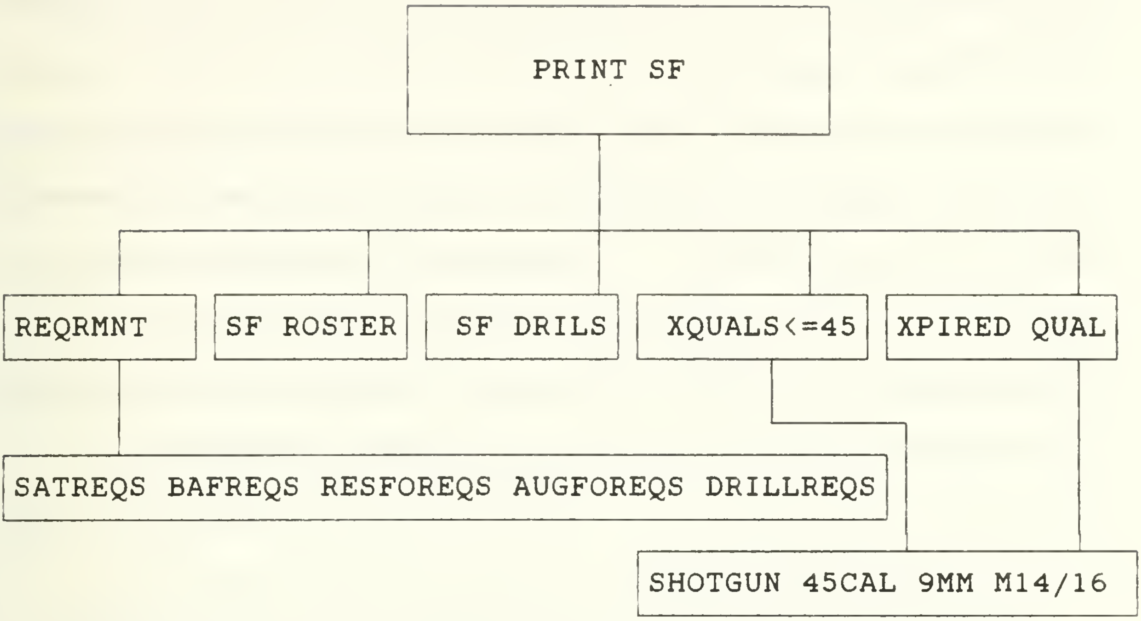
The second option in the DRILLS pop up menu, EDIT DRILLS, allows the user the ability to either edit a past drill, or add a new drill. This option is necessary for the Security Officer and actually the duty Weapons Officer when the daily or weekly drills are completed. The selection on the menu is made, and a form is presented to the user in which specific detailed information is filled in. This information now becomes a part of the chronological history of the DRILLS database. This action is accomplished by mapping the internal dBASE IV software through a form and into the database file (drills.dbf). If the user intends to edit the information on a specific drill, he simply selects this option, finds the drill he wishes to edit, and does so. The changes are then saved and written to the database file.

c. Print Drills

The third and final option on the DRILLS pop up menu, PRINT DRILLS, is provided so that the user may again have hard copy data for physical filing, or requirements from up the chain of command. This action is accomplished in the internal software, by mapping the option to the database file (drills.dbf) and sending a message to the printer to print the files. Currently this option is designed to print all files located in the database, i.e. all drills that have been logged into the system will print out in a convenient tabular format. The recommendation here is that the Security Officer keep only the drills for the current year in the database, all others can be archived. All drill information should be saved for training and reference, but within the system itself a year's worth at a time is sufficient.

4. Print Security Force

Returning to the main menu of the Security Force System, we will discuss the selection of the fourth option, PRINT SF. This is an acronym for Print Security Force. This option is an extra tool added to allow the user a more convenient access to the print command. A layered menu diagram is provided below.



Upon selection of this option, the user is presented with another pop up menu (PRINTSF) that will allow the user to print requirements, rosters, drills, a list of personnel who are within 45 days of small arms qualification expiration, and a list of personnel who's small arms qualifications have already expired. A sample menu diagram is provided below.

```
PRINT REQUIREMENTS   ->
PRINT ROSTER
PRINT DRILL REPORT
PRINT XQUALS <=45    ->
PRINT EXPIRED QUALS  ->
```

a. Print Requirements

The first of these options in the PRINTSF pop up menu, PRINT REQUIREMENTS, allows the user to print specific requirements as it did with the same option in the REQSSF pop up menu we previously reviewed. The arrow, once again, indicates to the user that there is an additional horizontal bar menu (REQSALL) attached for further options to clarify choice of requirements. Refer to the REQSALL menu to review these additional menu options. This action is accomplished in entirely the same fashion as with the PRINT REQUIREMENTS function in the pop up menu (REQSSF) as previously discussed.

b. Print Roster

The second option on the PRINTSF pop up menu, PRINT ROSTER, allows the user to print the security force roster in the same manner in which this option did in the ROSTER pop up menu. The absence of an arrow indicates to the user that there are no further layered menu selections. When this option is selected, the internal software maps the print command through the database file and sends the information to the printer. The Security Force Roster is now available in a hard copy format.

c. Print Drill Report

The third option in the PRINTSF pop up menu, PRINT DRILL REPORT, allows the user to print the information stored in the database files on drill reports. This action is accomplished in the exactly as in the Drills pop up menu. The selection is made, and the internal software maps the print command through the database file and outputs the data to the printer. The first three options in the PRINTSF pop up menu are provided to the user for convenience. If the user enters the Security Force System with the sole intent of printing out some information, it is much more convenient to use the PRINTSF pop up menu selections, than to navigate through several menus to accomplish the same task. The final two options on the PRINTSF pop up menu introduce two new ideas, and are presented below.

d. Print Qualifications Nearing Expiration

The fourth option in the PRINTSF pop up menu, PRINT XQUALS < = 45, is an acronym for Print those personnel whose small arms qualifications are within 45 days of expiration. This option allows the user to have a hard copy listing of personnel who must be considered for range qualifications in the very near future. It is a good planning strategy to be able to know in advance which personnel need to be requalified on the weapons that they will be using in a security alert situation. This option is also marked with an arrow that lets the user know there are further menu options available. The menu attached to this option is a horizontal bar menu (PEXPIRE) that offers the user to choose the type of weapons qualification to check. A sample menu diagram is provided below.

```
SHOTGUN   45CAL   9MM   M14/16
```

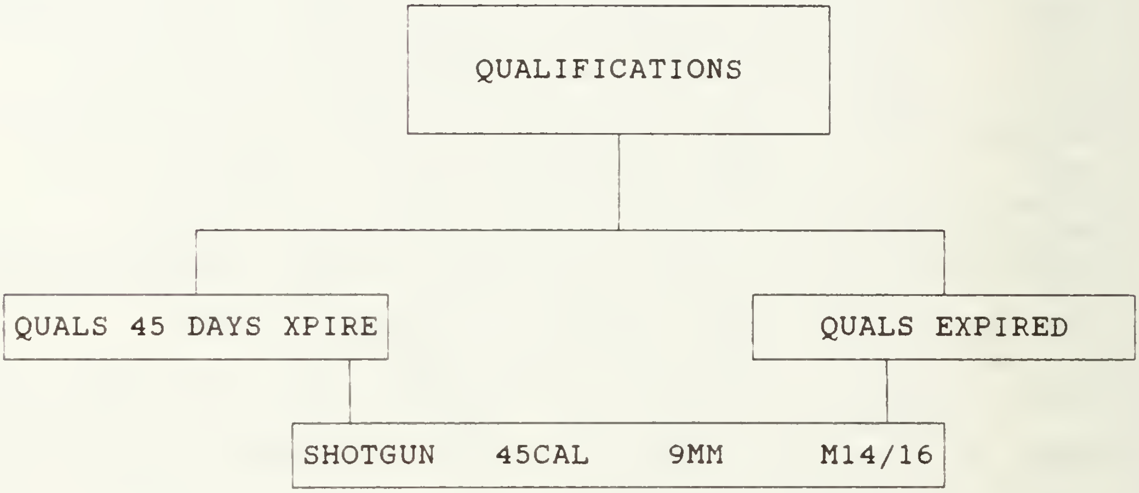
The first option in the PEXPIRE bar menu, SHOTGUN, allows the user to check all personnel qualifications on the shotgun. This is accomplished through the internal software screening the information in the database (mem_info.dbf) with the use of a filter that specifies expiration dates by adding 365 days to the qualification dates. The filter checks the dates by subtracting 365 days from today's date and further calculating to see if the date of qualification is within 45 days of this calculated number. If it is, the file is flagged and sent to the printer. The remaining three options in this menu are all accomplished in precisely the same manner as the SHOTGUN option was done. Within the database file (mem_info.dbf), information is stored on qualification dates for the four separate weapons listed above. Accomplishment of this menu feature is done through filters that check and calculate these individual qualification dates.

e. Print Expired Results

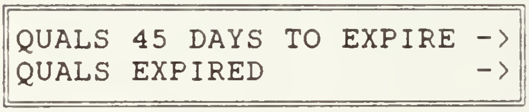
The next option in the PRINTSF pop up menu, PRINT EXPIRED QUALS, allows the user to print a list of those personnel whose qualifications have already expired. This is a very important tool because it is mandated by higher authority that all personnel who handle weapons must be qualified on the weapon they are assigned. Having qualifications expire on a Security Force Member would technically preclude that person from handling a weapon. This of course makes the member virtually useless to the Security Force, at least until the member can be requalified. A list of this type is useful to post in the weapons locker to remind duty weapons personnel not to issue weapons to unqualified Security Force members. This could be a very serious problem if an unqualified person accidentally discharged a firearm during a drill injuring or worse yet, inadvertently killing a shipmate. This option is also marked with an arrow which signals the user that there are other menu options available. The horizontal bar menu attached (PEXIRED) allows the user to once again choose the weapon in which qualifications verification is needed. This horizontal bar menu is identical to the menu above (PEXPIRE). The functions it provides are very similar in that each option screens the database for the appropriate weapons qualification date, adds 365 days to it, and compares that to today's date. If the date of today is greater than the calculated date, then the qualification is expired. This file is then flagged and sent to the printer. Once again, the majority of options provided in the PRINTSF pop up menu are all duplications of print commands that are found in other menus throughout the Security Force System. They are provided in one centralized location to simplify the user interface.

5. Qualifications

Returning to the main menu of the Security Force System, the next option available to the user is QUALIFICATIONS. This option provides the user a way to check on the small arms qualifications of the Security Force. A layered menu diagram is provide below.

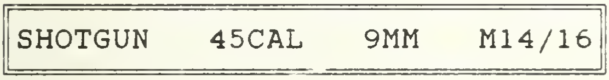


When this option is selected, a pop up menu (QUALS) is presented. A menu diagram is provided below.



a. List Qualifications Near Expiration

The first option available to the user, QUALS 45 DAYS TO EXPIRE, allows the user to view a list of personnel who are within 45 days of having a particular qualification expire. The arrow attached to this menu signals the user that further options are available. These options are presented in the form of another horizontal bar menu (EXPIRE) and allow the user to choose the type of weapon to be checked for qualifications expiration. A sample menu diagram is provided below.



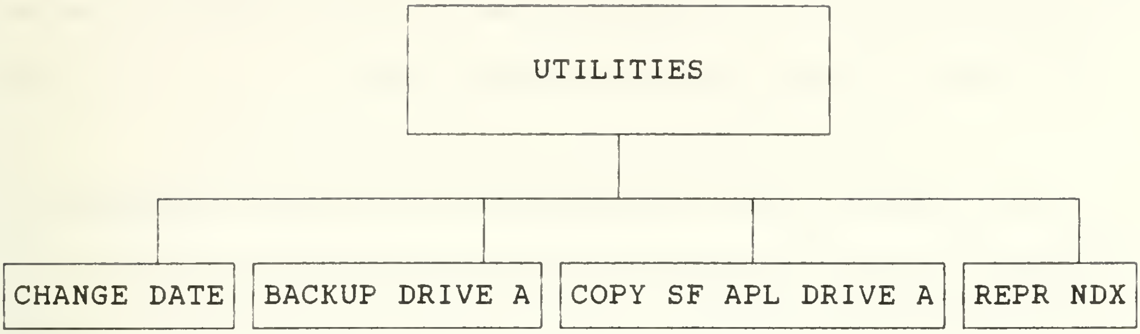
The functions in this menu are accomplished in a nearly identical format as described previously in the PRINTSF pop up menu section; however, there are general differences. The first option in the EXPIRE bar menu, SHOTGUN, allows the user to check all personnel qualifications on the shotgun. This is accomplished through the internal software screening the information in the database (mem_info.dbf) with the use of a filter that specifies expiration dates by adding 365 days to the qualification dates. The filter checks the dates by subtracting 365 days from today's date and further calculating to see if the date of qualification is within 45 days of this calculated number. If it is, the file is flagged and sent to the screen. The remaining three options in this menu are all accomplished in precisely the same manner as the SHOTGUN option was done. Within the database file (mem_info.dbf), information is stored on qualification dates for the four separate weapons listed above. Accomplishment of this menu feature is done through filters that check and calculate these individual qualification dates.

b. Expired Qualifications

The next option in the QUALS pop up menu, QUALS EXPIRED, allows the user to print a list of personnel whose qualifications have already expired. The arrow again indicates to the user the existence of an additional menu (EXPIRED) with options available to further specify areas of qualification. This information is nearly identical to the procedures discussed above.

6. Utilities

The sixth option in the main menu of the Security Force System, UTILITIES, allows the user to perform several hardware and system related functions. These functions include changing the system date, backing up the dbf file to the A drive, backing up the entire directory to the A drive, and repairing corrupted indexes. A layered menu diagram is provided below.



Upon selection of this menu option, the user will be presented with an additional pop up menu (UTILS). A menu diagram is provided below.

CHANGE SYSTEM DATE
BACKUP TO DRIVE A
COPY SF APPL DRIVE A
REPAIR INDEXES

a. Change System Date

The first option of the UTILS pop up menu, CHANGE SYSTEM DATE, allows the user to update the date of the computer. This is very important in general for timeliness in applications, but specifically in this application since the qualifications expiration procedures are all calculated based on system or current date. The risk factors involved with having the incorrect date on a shipboard computer are very real with computer down time for maintenance and power outages as discussed previously. This option allows the user to perform this function without having to exit the system to change the date in the DOS environment; hence it saves time that is better spent on other tasks.

b. Backup Drive A

The second option in the UTILS pop up menu, BACKUP TO DRIVE A, allows the user to copy the database files to a floppy disk for backup, etc. For usage as a backup system, the importance of this feature is critical. Loss of data in any environment equates to lost time as well as a real dollar loss. Shipboard applications are no exception. The gravity of the situation is compounded for a deployed ship without the luxury of borrowing a backup application; therefore, this feature when utilized will be invaluable in an operational environment. As a sharing device, this feature has other benefits. Time and dollars can also be saved if one can be convinced to view information via the floppy disk medium vice in hard copy (paper). Double density floppy disks can store approximately 360 k bytes of information. A single sheet of paper can store 2k to 4k bytes of information, so a double density floppy disk can store approximately the same amount of information as 90 to 100 sheets of paper. The average cost of 100 sheets of paper is approximately two dollars as compared to the cost of a double density floppy disk at about twenty five cents when purchased in bulk. The fact that floppy disks are reusable and more economical than paper results in large savings for the Navy.

c. Download Application

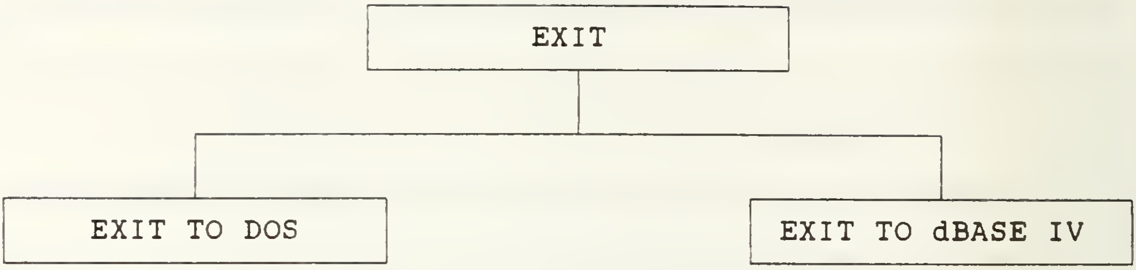
The third option in the UTILS menu, COPY SF APPL DRIVE A, allows the user to copy the entire directory to the A drive. This feature provides very similar benefits to the previous option as a backup. Instead of copying only the database information files, this option copies all files to the disk. The entire system can be reconstructed, if necessary, from the transfer of this information.

d. Repair Indexes

The fourth and final option in the UTILS pop up menu, REPAIR INDEXES, allows the user to repair corrupted indexes in the system which will cause incorrect presentation of data. Information designed to be presented in chronological order may not be presented as such. This option is a tool that can repair the problems that are inherently associated with power outages, fluctuations, and interruption.

7. Exit

The seventh and final option on the main menu of the Security Force System is the EXIT option. A layered menu diagram is provided below.



Selection of this option presents the user with a final pop up menu (EXIT). A menu diagram is presented below.

EXIT TO DOS
EXIT TO dBASE IV

a. Exit to dBASE IV Control Center

The second and final option in the EXIT pop up menu, EXIT TO dBASE IV, allows the user to exit the Security Force System and return to the dBASE IV control center. This is a helpful option if the user has programmed additional applications into dBASE IV for database related file functions in other areas of his or her work. An example of this would be utilizing the Security Force System and the PRP Database System without the benefit of the PRP Expert System.

The system described in detail above is currently coded and operational. Its intended use is to assist the Security Officer, the Weapons Officer and the PRP Officer with the administrative burdens inherent with these programs.

b. Exit to Dos

The first option in the EXIT pop up menu, EXIT TO DOS, allows the user to exit the system and is returned into the DOS environment.

IV. IMPLEMENTATION CONSIDERATIONS

A. SHIP'S SECURITY FORCE SYSTEM AS AN NWMS SUBMODULE

The Ship's Security Force System is designed as a submodule of the Nuclear Weapons Management System. The project can be a stand alone application as well, and has been designed to provide this capability. It is recommended that the system be introduced to the fleet, at the present time, as a prototype. Prototyping is rapidly becoming the software engineering "weapon" of choice, and the Navy could benefit greatly from its step-by-step refinement process. A software package must be tested rigorously before it can be depended upon as a primary tool. Prototyping allows the user to become a part of the testing process. Implementation of this combined system package could begin immediately, and it is recommended that it first be implemented in either a shore based facility such as NUCWEPTRAGRU, or a small ship afloat. Costs incurred by the Navy for this program will include the price of one copy of the VP-Expert software package per ship or station. dBASE III+ and dBASE IV currently exist in the fleet, hence no further costs are anticipated. The PRP program is in need of a tool like the Nuclear Weapons Management System, and this package could be an introduction to more serious software related initiatives in the fleet.

B. SHIP'S SECURITY FORCE SYSTEM AS A STAND ALONE APPLICATION

The SSFS System in a stand alone application can benefit the PRP officer, the Weapons Officer and the Security Officer alike. It is recommended that this application be implemented as a prototype to the fleet. With the popularity and availability of dBASE III+ and dBASE IV in the Navy, and the references and materials available, this application can be utilized to assist in administrative tasks of any ship or station. The flexibility introduced in the system to allow for nuclear capable as well as non-nuclear capable shipboard application is another of the versatile elements of the system and of the dBASE IV software package. Costs incurred are minimal since most ships in the Navy today have at least one copy of dBASE IV aboard.

C. FUTURE EXPANSION POSSIBILITIES

The expansion possibilities for this system are widespread and realistic. With the advent of the CIM initiative and the Desktop III and IV contracts, the government is propelling the Armed Forces into the 21st century. Computer application is not the future, it is the present. The U.S. NAVY must keep abreast of the advancements in Information Technology if we are to survive.

Future expansion modules for this system can include the following: NWMS modules to assist with the administrative tasks associated with Technical Operations and Training in the Nuclear Weapons Management Programs.

A dBASE IV application that allows the Damage Control Assistant (DCA) to track training, material readiness and safety in relation to the DCA's responsibilities in an Accident Incident Nuclear related drill. Damage control personnel must be trained and qualified in their respective job related tasks. The subsequent administrative requirements are similar to those of security force personnel. A stand alone system for the DCA is a realistic possibility in the training and tracking of other non-nuclear related damage control areas such as fire fighting, flooding, and emergency hull repair.

Another expansion possibility is a submodule for the ASW or Special Weapons Officer to track training and other administrative functions for personnel responsible for maintenance on special weapons. Preventative maintenance and training are critical in the Navy, and computer automated systems can have been accomplishing similar tasks for years.

Another application for the system is assisting the Weapons Officer with the administrative requirements involved with handling ammunition. Assistance with the administrative burden can be provided in the form of a submodule in the NWMS system.

Two additional uses of this system are envisioned in the form of expert system applications. The first is Special Incidents reporting, and the second is the CASREP system. Both of these systems offer written flow charts of required procedures and reports that must be submitted based on current directives. As with the expert system application in the NWMS, administrative burdens and time intensive publications searches can be avoided. These are but a few of the possibilities that are available when working with a combination of a database and expert systems.

V. CONCLUSIONS AND RECOMMENDATIONS

This thesis examines the need for an automated system to aid in accomplishing administrative duties in the Nuclear Weapons Management and related security programs. As stated throughout the thesis, there is a need for PRP applications, and applications to assist in other areas related to the Nuclear Weapons Management programs. The Nuclear Weapons Management System utilizes both dBASE IV and VP-Expert to support the accomplishment of required administrative duties. The Ship's Security Force System utilizes dBASE IV to demonstrate administrative effectiveness in a non-nuclear application.

Based on the experiment accomplished in this thesis, it is recommended that the system be distributed to several controlled locations. Formulate a team of "experts" to study the use of the software and any associated problems. The team can spend some time updating the software and returning it to these same controlled locations. After several iterations, the software will be evaluated for release to the fleet. The hardware, software and expertise are available in the Navy today.

Secondly, with regards to security, advances in hardware are so rapid that it is difficult to keep abreast of the latest technology. Currently, there exists a portable/removable hard drive available in the marketplace. This can increase security of sensitive information without having to employ expensive encryption techniques and other security features. After the PRP Officer accomplishes the required administrative duties, the hard drive is removed and placed in the Officers safe.

With hardware prices declining, software prices rising, and the Navy budget under increasing scrutinization, the importance of user generated software packages like these will continue to be realized.

APPENDIX A - USER'S MANUAL

I. INSTALLATION PROCEDURES

A. NWMS BATCH FILES

The following table provides a list of the NWMS batch files.

FILE NAME	FUNCTION
1.BAT	Load and run VP-Expert
2.BAT	Load and run dBase IV for SSFS
3.BAT	Load and display Administrative Tools Menu
4.BAT	Exit NWMS and return to DOS root directory
A.BAT	Load and run dBase IV for PRP roster
B.BAT	Load and print PRP questionnaire text file
C.BAT	Return to NWMS Main Menu
MAINMENU.SCR	Main Menu display
ADMNMENU.SCR	Administrative Tools Menu display
NWMS.BAT	Load and display NWMS Main Menu
PRPQUES.TXT	PRP Questionnaire text file

It is assumed that the system will be installed onto a hard disk that has been designated as a "C" drive. It is also assumed that the source files are originating from floppy diskettes in the "A" drive. If a higher letter hard drive can be used instead of "C", just substitute the appropriate letter. Additionally, a "B" drive can be used instead of the "A" source drive, just substitute the appropriate letter.

To install the NWMS files, use the following steps:

1. Establish a directory called "NWMS" with the command:
"MD NWMS" at the C> prompt.

2. The NWMS batch files will be located on the provided source diskette under the NWMS subdirectory. Change to this subdirectory with the command: "A:\NWMS".

3. Transfer all files from A:\NWMS to C:\NWMS with the following command: "COPY *.* C:\NWMS".

The NWMS batch files are now installed. To run the NWMS program, enter the NWMS subdirectory on the hard drive with the command: "CD C:\NWMS" and then type NWMS. The main menu should appear. It will be necessary to ensure that dBASE IV and VP-Expert are installed before the menu will be useful, but the initial steps of the installation have been completed.

B. SSFS SYSTEM

Before SSFS can be installed, dBASE IV must already exist on the C drive (hard disk) under the directory name "DBASE". If this is not the case, follow the steps that follow.

1. Install dBASE IV, if it has not already been installed, and establish it under the directory name DBASE.

2. If the program has already been installed but exists under a different directory name, an adjustment must be made. The directory name for dBASE IV must be renamed to DBASE or the batch file that executes dBASE IV (2.BAT) must be modified to refer to the actual directory name. Either action is easily accomplished by using utility software (for example, PC Tools). To modify 2.BAT, use the text editor and substitute the actual directory name for dBASE IV on the first line. Save the change and exit the text editor. To change the

directory name, locate the directory tool that renames directories, and change the directory name to DBASE.

Once dBASE IV is correctly established, install the SSFS system by copying the files from the source diskette to the DBASE directory on the C drive. The following steps apply:

1. Insert the diskette into the A drive and establish an A> prompt with the command "A:".

2. Change to the subdirectory DBASE on the A drive with the DOS command "CD\DBASE".

3. Copy the SSFS files to dBASE IV with the command "COPY *.* C:\DBASE".

The installation of the SSFS system is complete.

C. PRP EXPERT

For installation procedures of the PRP Expert system, refer to [Ref. 1].

II. GETTING STARTED

A. NWMS

To start the NWMS program, change to the NWMS directory with the command "CD\NWMS". Next, begin the program by typing "NWMS" and pressing ENTER. The main menu should be visible. To make a selection from the menu, type the appropriate number and press ENTER. To exit the menu, select "Return to DOS" and press ENTER. The user will be returned to the C drive root directory.

B. SSFS SYSTEM

To enter the SSFS system, select SSFS from the main menu. This will bring the user directly into dBASE IV and present the sign on banner for the SSFS system. A detailed description of menu options is available in chapter III of the thesis.

C. PRP EXPERT

Refer to [Ref. 1].

APPENDIX B - REFERENCES / BIBLIOGRAPHY

[1] Rodgers, Daniel J., Design and Implementation of a Nuclear Weapons Management System Submodule: Personnel Reliability Program Expert System, Naval Postgraduate School, Monterey, Ca., 1991.

[2] Nuclear Safety Administration for COMNAVSURFPAC Units, K-644-9031, Nuclear Weapons Training Group Pacific, San Diego, Ca., August 1985.

[3] Department of Defense Directive 5210.42, Nuclear Weapon Personnel Reliability Program, Washington, D.C., 6 December 1985.

[4] Lind, Howard R., Improving Shipboard Nuclear Weapons Security Through Computer Automation of the Personnel Reliability Program's [PRP] Administrative Requirements, Naval Postgraduate School, Monterey, Ca., 1988.

[5] Turban E., Decision Support and Expert Systems, Macmillan, Second Edition, 1990.

[6] Simpson A., dBase III Plus Programmer's Reference Guide, Sybex, Alameda, Ca., 1987.

[7] Simpson A., Understanding dBASE IV, Sybex, Alameda, Ca., 1989.

[8] Kroenke D., Dolan K., Database Processing, Macmillan, Third Edition, 1988.

APPENDIX C - SSFS DATA DICTIONARY

DATABASE FILES (.DBF)

1. Name: MEM_INFO
Narrative Description: Central NWMS database which contains information describing each member of PRP along with other ship's force members who are involved with the security force.

Representation:

Field Name	type	width
lastname	char	12
initials	char	4
rank_rate	char	5
ssn	char	11
department	char	4
division	char	4
reportdate	date	8
eaos	date	8
sec_class	char	3
shtgn_qual	logical	1
shtgn_date	date	8
F45_qual	logical	1
F45_date	date	8
F9mm_qual	logical	1
F9mm_date	date	8
M14_16qual	logical	1
M14_16date	date	8
billet	logical	1
billetype	char	10
qual	char	7

Unique identifier (key field): ssn
Note: The logical field "billet" identifies whether or not the record is a PRP member.

2. Name: DRILLS
Narrative description: Database file used to store information on security force drills.

Representation:

Field Name	type	width
date	date	8

duty_sect	numeric	1
location	character	30
type	character	6
scenario	character	254
comments	character	254

Unique identifier (key field): date

Note: The field "date" is important for a chronological listing of security force drills.

3. Name: REQUIREM

Narrative description: Database file used to store information on security force requirements.

Representation:

Field Name	type	width
sat	memo	10
baf	memo	10
resfor	memo	10
augfor	memo	10
sf_drills	memo	10

Unique identifier (key field): memo

Note: The field "memo" is important because it provides a text representation to supply the user with reference materials for security force teams and drills.

APPENDIX D - PROGRAM CODE

```
****************************************************************
**********
* Program......: MEM_INFO.PRG
* Author.......: Sidney R. Settlemyer, Lt. USN
* Date.........: 7-23-91
* Notice.......: Written as partial fulfillment for thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: THIS PROGRAM LISTS AND UPDATES SECURITY
FORCE PERSONNEL

* Description..: Main routine for menu system
****************************************************************
**********

*-- Setup environment
SET CONSOLE OFF
IF TYPE("gn_ApGen")="U"
   CLEAR WINDOWS
   CLEAR ALL
   CLOSE ALL
   CLOSE PROCEDURE
   gn_ApGen=1
ELSE
   gn_ApGen=gn_ApGen+1
   IF gn_ApGen > 4
      Do Pause WITH "Maximum level of Application nesting
exceeded."
      RETURN
   ENDIF
   PRIVATE gn_oldsize
   gn_oldsize=gn_scrsize
   PRIVATE gc_bell, gc_carry, gc_clock, gc_century,
gc_confirm, gc_deli,;
           gc_safety, gc_status, gc_score, gc_talk, gc_key,
gc_prognum,;
           gc_quit, gc_color, gc_display, gl_color,
gl_batch, gn_scrsize
ENDIF
*-- Store some sets to variables
gc_bell   =SET("BELL")
gc_carry  =SET("CARRY")
gc_clock  =SET("CLOCK")
gc_color  =SET("ATTRIBUTE")
gc_century=SET("CENTURY")
gc_confirm=SET("CONFIRM")
gc_cursor =SET("CURSOR")
gc_deli   =SET("DELIMITERS")
gc_display=SET("DISPLAY")
```

```
gc_safety =SET("SAFETY")
gc_status =SET("STATUS")
gc_score  =SET("SCOREBOARD")
gc_talk   =SET("TALK")
SET CONSOLE ON
IF gc_display <> "EGA25"
   gn_error=0
   ON ERROR ??
   SET DISPLAY TO EGA25
   ON ERROR
ENDIF

SET BELL ON
SET CARRY OFF
SET CENTURY OFF
SET CLOCK OFF
SET CONFIRM OFF
SET DELIMITERS TO ""
SET DELIMITERS OFF
SET DEVICE TO SCREEN
SET ESCAPE ON
SET EXCLUSIVE OFF
SET LOCK ON
SET MESSAGE TO ""
SET PRINT OFF
SET REPROCESS TO 4
SET SAFETY ON
SET TALK OFF

*-- Initialize global variables
gl_batch=.F.        && is a batch operation in progress
gl_color= ISCOLOR() .AND. SET("DISPLAY") <> "CGAMONO"
gn_error=0          && 0 if no error, otherwise an error
occurred
gn_ikey=0           && keypress returned from the INKEY()
function
gn_scrsize=21       && number of lines on screen
gn_send=0           && return value from popup of position
menus
gn_trace=1          && sets trace level, however you need to
change template
gc_brdr='1'         && border to use when drawing boxes
gc_dev='CON'        && Device to use for printing - See
Proc. PrintSet
gc_key='N'          && leave the application
gc_prognum='  '     && internal program counter to handle
nested menus
gc_quit=' '         && memvar for return to caller
listval='NO_FIELD'  && Pick List value

*-- remove asterisk to turn clock on
```

```
* SET CLOCK TO

*-- Blank the screen
SET COLOR TO
CLEAR
SET SCOREBOARD OFF
SET STATUS OFF

*-- Define menus
DO MPDEF            && execute Menu Process DEFinition

*-- Execute main menu
DO WHILE gc_key = 'N'
   DO MEMMAIN WITH "BOO"
   IF gc_quit = 'Q'
      EXIT
   ENDIF
   ACTIVATE WINDOW Exit_App
   lc_conf=SET("CONFIRM")
   lc_deli=SET("DELIMITER")
   SET CONFIRM OFF
   SET DELIMITER OFF
   @ 1,2 SAY "Do you want to leave this application?" ;
         GET gc_key PICT "!" VALID gc_key $ "NY"
   READ
   SET CONFIRM &lc_conf.
   SET DELIMITER &lc_deli.
   RELEASE lc_conf, lc_deli
   DEACTIVATE WINDOW Exit_App
ENDDO

*-- Reset environment
DEACTIVATE WINDOW FullScr
?? Color(gc_color)
gn_ApGen=gn_ApGen-1
SET BELL  &gc_bell.
SET CARRY &gc_carry.
SET CLOCK &gc_clock.
SET CENTURY &gc_century.
SET CONFIRM &gc_confirm.
SET CURSOR  &gc_cursor.
SET DELIMITERS &gc_deli.
SET DISPLAY TO &gc_display.
SET STATUS &gc_status.
SET SAFETY &gc_safety.
SET SCORE  &gc_score.
SET TALK   &gc_talk.

IF gn_Apgen < 1
   ON KEY LABEL F1
   CLEAR WINDOWS
   CLEAR ALL
   CLOSE ALL
   CLOSE PROCEDURE
   SET ESCAPE ON
   SET MESSAGE TO ""
   CLEAR
ELSE
   DEFINE WINDOW FullScr FROM 0,0 TO gn_oldsize+3,79 NONE
   DEFINE WINDOW Savescr FROM 0,0 TO gn_oldsize,79 NONE
   DEFINE WINDOW Helpscr FROM 0,0 TO gn_oldsize,79 NONE
   ACTIVATE WINDOW FullScr
ENDIF
RETURN

************************************************************
*******************
* Description..: Procedure files for generated menu system.
* The programs that follow are common to main routines
* The last procedure is the Menu Process DEFinition
************************************************************
*******************
PROCEDURE Lockit
PARAMETER ltype
IF NETWORK()
   gn_error=0
   ON ERROR DO Multerr
   IF ltype = "1"
     ll_lock=FLOCK()
   ENDIF
   IF ltype = "2"
     ll_lock=RLOCK()
   ENDIF
   ON ERROR
ENDIF
RETURN

PROCEDURE Info_Box
PARAMETERS lc_say
? lc_say
? REPLICATE("-",LEN(lc_say))
?
RETURN
* EOP: Info_Box

PROCEDURE get_sele
*-- Get the user selection & store BAR into variable
gn_send = BAR()  && Variable for print testing
DEACTIVATE POPUP
RETURN

PROCEDURE ShowPick
```

```
listval=PROMPT()
IF LEFT(entryflg,1)="B"
   lc_file=POPUP()
   DO &lc_file. WITH "A"
   RETURN
ENDIF
IF TYPE("lc_window")="U"
   ACTIVATE WINDOW ShowPick
ELSE
   ACTIVATE WINDOW &lc_window.
ENDIF
STORE 0 TO ln_ikey,x1,x2
ln_ikey=LASTKEY()
IF ln_ikey=13
   x1=AT(TRIM(listval)+',',lc_fldlst)
   IF x1 = 0
      lc_fldlst=lc_fldlst+TRIM(listval)+','
   ELSE
      x2=AT(',',SUBSTR(lc_fldlst,x1))
      lc_fldlst=STUFF(lc_fldlst,x1,x2,'')
   ENDIF
   CLEAR
   ? lc_fldlst
ENDIF
ACTIVATE SCREEN
RETURN
* EOP: ShowPick

PROCEDURE Cleanup
*-- test whether report option was selected
DO CASE
CASE gc_dev='CON'
   ? "  Press any key to continue..."
   xx=INKEY(0)
CASE gc_dev='PRN'
   SET PRINT OFF
   SET PRINTER TO
CASE gc_dev='TXT'
   CLOSE ALTERNATE
ENDCASE
gc_dev='CON'
RETURN

* EOP: Cleanup

PROCEDURE Pause
PARAMETER lc_msg
*-- Parameters : lc_msg = message line
IF TYPE("lc_message")="U"
   gn_error=ERROR()
ENDIF
```

```
lc_msg = lc_msg
lc_option='0'
ACTIVATE WINDOW Pause
IF gn_error > 0
   IF TYPE("lc_message")="U"
      @ 0,1 SAY [An error has occurred !! - Error message:
]+MESSAGE()
   ELSE
      @ 0,1 SAY [Error # ]+lc_message
   ENDIF
ENDIF
@ 1,1 SAY lc_msg
WAIT " Press any key to continue..."
DEACTIVATE WINDOW Pause
RETURN

* EOP: Pause

PROCEDURE Multerr
*-- set the global error variable
gn_error=ERROR()
*-- contains error number to test
lc_erno=STR(ERROR(),3)+','
*-- option var.
lc_opt='T'
*-- Dialog box for options Try again and Return to menu.
IF lc_erno $ "108,109,128,129,"
   ACTIVATE WINDOW Pause
   @ 0,2 SAY lc_erno+" "+MESSAGE()
   @ 2,22 SAY "T = Try again, R = Return to menu." GET
lc_opt ;
PICTURE "!" VALID lc_opt $ "TR"
   READ
   DEACTIVATE WINDOW Pause
   IF lc_opt = "R"
      RETURN
   ENDIF
ENDIF
*-- Display message and return to menu.
IF .NOT. lc_erno $ "108,109,128,129,"
   DO PAUSE WITH ERROR()
   RETURN
ENDIF
*-- reset global variable
gn_error=0
*-- Try the command again
RETRY
RETURN

* EOP: Multerr
```

```
PROCEDURE Trace
*  Desc: Trace procedure - to let programmer know what
module
*            is about to execute and what module has
executed.
PARAMETERS p_msg, p_lvl
*-- Parameters : p_msg = message line, p_lvl = trace level
lc_msg = p_msg
ln_lvl = p_lvl
lc_trp = ' '
IF gn_trace < ln_lvl
   RETURN
ENDIF
DEFINE WINDOW trace FROM 5,0 TO 16,79 DOUBLE
ACTIVATE WINDOW trace
DO WHILE lc_trp <> 'Q'
   CLEAR
   @ 2,40-LEN(lc_msg)/2 SAY lc_msg
   @ 4,05 SAY 'S - Set trace level, D - Display status, M -
display Memory'
   @ 5,05 SAY 'P - Turn printer on, Q - to Quit'
   lc_trp = 'Q'
   @ 5,38 GET lc_trp PICTURE "!"
   READ
   DO CASE
   CASE lc_trp = 'S'
      @ 2,01 CLEAR
      @ 2,33 SAY 'Set trace level'
      @ 4,05 SAY 'Enter trace level to change to:' GET
gn_trace PICTURE '#'
      @ 5,05 SAY '                 '
      READ
      IF gn_trace=0
         @ 2,01 CLEAR
         @ 3,05 SAY 'Trace is now turned off..To reactivate
Trace - Press [F3]'
         @ 4,05 say 'Press any key to continue...'
         WAIT ''
         ON KEY LABEL F3 gn_trace = 1
      ENDIF
   CASE lc_trp = 'D'
      DISPLAY STATUS
      WAIT
   CASE lc_trp = 'M'
      DISPLAY MEMORY
      WAIT
   CASE lc_trp = 'P'
      SET PRINT ON
   ENDCASE
ENDDO
SET PRINT OFF
```

```
RELEASE WINDOW trace
@ 24,79 SAY " "
RETURN

* EOP: Trace

PROCEDURE PrintSet
*-- Initialize variables
gc_dev='CON'
lc_choice=' '
gn_pkey=0
gn_send=3

DEFINE WINDOW printemp FROM 08,25 TO 17,56

DEFINE POPUP SavePrin FROM 10,40
DEFINE BAR 1 OF SavePrin PROMPT " Send output to ..." SKIP
DEFINE BAR 2 OF SavePrin PROMPT REPLICATE(CHR(196),24) SKIP
DEFINE BAR 3 OF SavePrin PROMPT " CON:   Console" MESSAGE
"Send output to Screen"
DEFINE BAR 4 OF SavePrin PROMPT " LPT1:  Parallel port 1 "
MESSAGE "Send output to LPT1:"
DEFINE BAR 5 OF SavePrin PROMPT " LPT2:  Parallel port 2"
MESSAGE "Send output to LPT2:"
DEFINE BAR 6 OF SavePrin PROMPT " COM1:  Serial port 1"
MESSAGE "Send output to COM1:"
DEFINE BAR 7 OF SavePrin PROMPT " FILE = REPORT.TXT" MESSAGE
"Send output to File Report.txt"
ON SELECTION POPUP SavePrin DO get_sele

ACTIVATE POPUP SavePrin
RELEASE POPUP SavePrin

gn_pkey=LASTKEY()
IF gn_send = 7
   gc_dev = 'TXT'
   SET ALTERNATE TO REPORT.TXT
   SET ALTERNATE ON
ELSE
   IF .NOT. (gn_send = 3 .OR. LASTKEY() = 27)
      gc_dev = 'PRN'
      temp = SUBSTR("   LPT1LPT2COM1 ",((gn_send-2)-1)*4,4)
      ON ERROR DO prntrtry
      SET PRINTER TO &temp.
      IF gn_pkey <> 27
         SET PRINT ON
      ENDIF
      ON ERROR
   ENDIF
ENDIF
RELEASE WINDOW printemp
```

```
RETURN

PROCEDURE prntrtry
PRIVATE lc_escape
lc_escape = SET("ESCAPE")
IF .NOT. PRINTSTATUS()
   IF lc_escape = "ON"
      SET ESCAPE OFF
   ENDIF
   gn_pkey = 0
   ACTIVATE WINDOW printemp
   @ 1,0 SAY "Please ready your printer or"
   @ 2,0 SAY "     press ESC to cancel"
   DO WHILE ( .NOT. PRINTSTATUS()) .AND. gn_pkey <> 27
      gn_pkey = INKEY()
   ENDDO
   DEACTIVATE WINDOW printemp
   SET ESCAPE &lc_escape.
   IF gn_pkey <> 27
      RETRY
   ENDIF
ENDIF
RETURN

* EOP: PrintSet

PROCEDURE Position
IF LEN(DBF()) = 0
   DO Pause WITH "Database not in use. "
   RETURN
ENDIF
SET SPACE ON
SET DELIMITERS OFF
ln_type=0           && sublevel selection
ln_rkey=READKEY()   && test for ESC or Return
ln_rec=RECNO()      && DBF record number
ln_num=0            && for input of a number
ld_date=DATE()      && for input of a date
lc_option='0'       && main option ie. Seek, Goto and Locate
*-- Scope ie. ALL, REST, NEXT <n>
STORE SPACE(10) TO lc_scp
*-- 1 = Character SEEK, 2 = For clause, 3 = While clause
STORE SPACE(40) TO lc_ln1, lc_ln2, lc_ln3
lc_temp=""
@ 0,00 SAY "Index order: "+IIF(""=ORDER(),"Database is in
natural order",ORDER())
@ 1,00 SAY "Listed below are the first 16 fields."
lc_temp=REPLICATE(CHR(196),19)
@ 2,0 SAY
CHR(218)+lc_temp+CHR(194)+lc_temp+CHR(194)+lc_temp+CHR(194)+
lc_temp
```

```
ln_num=240
DO WHILE ln_num < 560
   lc_temp=FIELD( (ln_num-240)/20 +1)
   @ (ln_num/80),MOD(ln_num,80) SAY CHR(179)+;
lc_temp+SPACE(11-LEN(lc_temp))+;
SUBSTR("= Char  = Date  = Logic = Num   = Float = Memo
  ",;
AT(TYPE(lc_temp),"CDLNFMU")*8-7,8)
   ln_num=ln_num+20
ENDDO
ln_num=1

DEFINE POPUP Posit1 FROM 8,30
DEFINE BAR 1 OF Posit1 PROMPT " Position by " SKIP
DEFINE BAR 2 OF Posit1 PROMPT REPLICATE(CHR(196),15) SKIP
DEFINE BAR 3 OF Posit1 PROMPT " SEEK Record" MESSAGE "Search
on index key" SKIP FOR ""=ORDER()
DEFINE BAR 4 OF Posit1 PROMPT " GOTO Record" MESSAGE
"Position to specific record"
DEFINE BAR 5 OF Posit1 PROMPT " LOCATE Record " MESSAGE
"Locate record for condition"
DEFINE BAR 6 OF Posit1 PROMPT " Return" MESSAGE "Return
without positioning"
ON SELECTION POPUP Posit1 DO get_sele

SET CONFIRM ON
DO WHILE lc_option='0'
  ACTIVATE POPUP Posit1
  lc_option = ltrim(str(gn_send))  && for popup
   IF LASTKEY() = 27 .OR. lc_option="6"
      GOTO ln_rec
      EXIT
   ENDIF
   DO CASE
   CASE lc_option='3'
      *-- Seek
      IF LEN(NDX(1))=0 .AND. LEN(MDX(1))=0
         DO Pause WITH "Can't use this option - No index
files are open."
         LOOP
      ENDIF
      ln_type=1
      lc_ln1=SPACE(40)
      DEFINE WINDOW Posit2 FROM 8,19 TO 15,62 DOUBLE
      ACTIVATE WINDOW Posit2
      @ 1,1 SAY "Enter the type of expression:" GET ln_type
PICT "#" RANGE 1,3
      @ 2,1 SAY "(1=character, 2=numeric and 3=date.)"
      SET CURSOR ON
      READ
      SET CURSOR OFF
      IF .NOT. (READKEY() = 12 .OR. READKEY() = 268)
         SET CONFIRM ON
         @ 3,1 SAY "Enter the key expression to search for:"
         IF ln_type=3
            @ 4,1 GET ld_date PICT "@D"
         ELSE
            IF ln_type=2
               @ 4,1 GET ln_num PICT "##########"
            ELSE
               @ 4,1 GET lc_ln1
            ENDIF
         ENDIF
         SET CURSOR ON
         READ
         SET CURSOR OFF
         SET CONFIRM OFF
         IF .NOT. (READKEY() = 12 .OR. READKEY() = 268)

lc_temp=IIF(ln_type=1,"TRIM(lc_ln1)",IIF(ln_type=2,"ln_num",
"ld_date"))
            SEEK &lc_temp.
         ENDIF
      ENDIF
      RELEASE WINDOWS Posit2
   CASE lc_option='4'
      *-- Goto
      ln_type=1
      DEFINE POPUP Posit2 FROM 8,30
      DEFINE BAR 1 OF Posit2 PROMPT " GOTO:" SKIP
      DEFINE BAR 2 OF Posit2 PROMPT REPLICATE(CHR(196),10)
SKIP
      DEFINE BAR 3 OF Posit2 PROMPT " TOP" MESSAGE "GOTO Top
of File"
      DEFINE BAR 4 OF Posit2 PROMPT " BOTTOM" MESSAGE "GOTO
Bottom of File"
      DEFINE BAR 5 OF Posit2 PROMPT " Record # " MESSAGE
"GOTO A Specific Record"
      ON SELECTION POPUP Posit2 DO get_sele
      ACTIVATE POPUP posit2
      ln_type = gn_send
      IF LASTKEY() <> 27
         IF ln_type=5
            DEFINE WINDOW Posit2 FROM 8,26 TO 13,50 DOUBLE
            ACTIVATE WINDOW Posit2
            ln_num=0
            @ 3,1 SAY "Max. Record # =
"+LTRIM(STR(RECCOUNT()))
            @ 1,1 SAY "Record to GOTO" GET ln_num PICT
"######" RANGE 1,RECCOUNT()
            SET CURSOR ON
            READ
            SET CURSOR OFF
            IF .NOT. (READKEY() = 12 .OR. READKEY() = 268)
               GOTO ln_num
            ENDIF
            RELEASE WINDOWS Posit2
         ELSE
           lc_temp=IIF(ln_type=3,"TOP","BOTTOM")
           GOTO &lc_temp.
         ENDIF
      ENDIF
   CASE lc_option='5'
      *-- Locate
      DEFINE WINDOW Posit2 FROM 8,16 TO 14,66 DOUBLE
      ACTIVATE WINDOW Posit2
      @ 1,19 SAY "ie. ALL, NEXT <n>, and REST"
      @ 1,01 SAY "Scope:" GET lc_scp
      @ 2,01 SAY "For:  " GET lc_ln2
      @ 3,01 SAY "While:" GET lc_ln3
      SET CURSOR ON
      READ
      SET CURSOR OFF
      IF .NOT. (READKEY() = 12 .OR. READKEY() = 268)
         lc_temp=TRIM(lc_scp)
         lc_temp=lc_temp + IIF(LEN(TRIM(lc_ln2)) > 0," FOR
"+TRIM(lc_ln2),"")
         lc_temp=lc_temp + IIF(LEN(TRIM(lc_ln3)) > 0," WHILE
"+TRIM(lc_ln3),"")
         IF LEN(lc_temp) > 0
            LOCATE &lc_temp.
         ELSE
            DO Pause WITH "All fields were blank."
         ENDIF
      ENDIF
      RELEASE WINDOW Posit2
   ENDCASE
   IF EOF()
      DO Pause WITH "Record not found."
      GOTO ln_rec
   ENDIF
   IF READKEY()=12 .OR. READKEY()= 268 .OR. LASTKEY()=27  &&
Esc was hit
      lc_option='0'
   ENDIF
ENDDO
SET CURSOR &gc_cursor.
SET DELIMITERS &gc_deli.
SET CONFIRM OFF
RETURN

* EOP: Position
```

```
PROCEDURE BefAct
SAVE SCREEN TO Browscr&lc_ApGen.
DEACTIVATE WINDOW Fullscr
SET SCOREBOARD ON
RETURN
* EOP: BefAct

PROCEDURE AftAct
CLEAR
SET SCOREBOARD OFF
ACTIVATE WINDOW Fullscr
RESTORE SCREEN FROM Browscr&lc_ApGen.
RELEASE SCREEN Browscr&lc_ApGen.
RETURN
* EOP: AftAct

PROCEDURE Postnhlp
ln_getkey=INKEY()
DEFINE WINDOW Temphelp FROM 3,12 TO 19,67
ACTIVATE WINDOW Temphelp
DO CASE
CASE "SEEK" $ PROMPT()
*-- HELP SEEK
? " SEEK <exp>"
?
? " Evaluates a specified expression and attempts to"
? " find its value in the master index of the database"
? " file.  Returns a logical true (.T.) if the index"
? " key is found, and a logical false (.F.) if it is"
? " not found."
?
? " Ex: SEEK CTOD('11/03/87')  -  converts the"
? "     expression from character to date and"
? "     then searches for the value in the index"
?
CASE LEFT(LTRIM(PROMPT()),4) $ "GOTO TOP BOTT Reco"
*-- HELP GOTO
? " GO/GOTO BOTTOM/TOP [IN <alias>]"
? " or"
? " GO/GOTO [RECORD] <record number> [IN <alias>]"
? " or"
? " <record number>"
?
? " Positions the record pointer to a specified record"
? " or location in the active database file."
?
? "      TOP moves the pointer to the first record"
? "      BOTTOM moves the pointer to the last record"
?
? " Ex: 4  -  moves the record pointer to record 4"
?
```

```
CASE "LOCATE" $ PROMPT()
*-- HELP LOCATE
? " LOCATE FOR <condition> [<scope>]"
? "      [WHILE <condition>]"
?
? " Searches the active database file, sequentially,"
? " for the first record that meets the specified"
? " criteria.  The function FOUND() returns true (.T.)"
? " if LOCATE is successful."
?
? " Ex: LOCATE FOR Age = '25' NEXT 5"
? "     searches for the next five records"
? "     containing 25 in the Age field"
?
CASE "Return" $ PROMPT()
?
? " Return to action in progress, with or without"
? " positioning the record pointer."
ENDCASE
ln_getkey = INKEY(0)
DEACTIVATE WINDOW Temphelp
RELEASE WINDOW Temphelp
RETURN
* EOP: Postnhlp

FUNCTION Color
*------------------------------------------------------------
----------------
* Format:
* COLOR( <expC> )
*  <expC> = NORMAL, HIGHLIGHT, MESSAGES, TITLES, BOX,
INFORMATION, FIELDS
*         or a variable with all colors store in it
*  Ver: dBASE 1.1
*
* The COLOR() function either returns or sets colors
returned with the
* SET("attribute") setting
* If <expC> is a color string then null is returned
otherwise the color
* setting is returned for one of dBASE's color options
*
* See Also: SET("attribute")
*
*------------------------------------------------------------
----------------
PARAMETERS set_color
PRIVATE color_num, color_str, cnt

set_color = UPPER(set_color)
IF set_color = "COLOR"
```

```
  *- Return standard, enhanced, border colors only
  RETURN SUBSTR(SET("attr"),1, AT(" &", SET("attr")))
ENDIF

*- Declare array to parse color options from SET("attr")
PRIVATE color_
DECLARE color_[8]
*- Determine if user is restoring colors vs. saving colors
IF " &" $ set_color
  color_str = ","+set_color+","                      && Restore
color attributes
ELSE
  color_str = ","+SET("ATTRIBUTE")+","               && Save
color attributes
ENDIF

*-- Stuff array with individual color setting
color_str = STUFF(color_str, AT(" &", color_str), 4, ",")
cnt = 1
DO WHILE cnt <= 8
  color_str = SUBSTR(color_str, AT(",", color_str ) +1 )
  color_[cnt] = SUBSTR(color_str, 1, AT(",", color_str ) -
1)
  cnt = cnt + 1
ENDDO

IF " &" $ set_color
  *-- Set color back
  SET COLOR TO ,,&color_[3].                         && Border
color
  SET COLOR OF NORMAL TO &color_[1].
  SET COLOR OF HIGHLIGHT TO &color_[2].
  SET COLOR OF MESSAGES TO &color_[4].
  SET COLOR OF TITLES TO &color_[5].
  SET COLOR OF BOX TO &color_[6].
  SET COLOR OF INFORMATION TO &color_[7].
  SET COLOR OF FIELDS TO &color_[8].
ELSE
  *-- Return color string requested
  DO CASE
  CASE set_color $ "NORMAL"
    color_num =  1
  CASE set_color $ "HIGHLIGHT"
    color_num =  2
  CASE set_color $ "BORDER"
    color_num =  3
  CASE set_color $ "MESSAGES"
    color_num =  4
  CASE set_color $ "TITLES"
    color_num =  5
  CASE set_color $ "BOX"
```

```
    color_num =  6
  CASE set_color $ "INFORMATION"
    color_num =  7
  CASE set_color $ "FIELDS"
    color_num =  8
  ENDCASE
ENDIF
RETURN IIF(" &" $ set_color, "", color_[color_num])



*************************************************************
**********
* Program......: MPDEF
* Author.......: Sidney R. Settlemyer, Lt. USN
* Date.........: 7-23-91
* Notice.......:
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: THIS PROGRAM LISTS AND UPDATES SECURITY
FORCE PERSONNEL

* Description..: Defines all menus in the system
*************************************************************
**********
PROCEDURE MPDEF
IF gl_color
   SET COLOR OF NORMAL TO W+/B
   SET COLOR OF MESSAGES TO W+/N
   SET COLOR OF TITLES TO W/B
   SET COLOR OF HIGHLIGHT TO RG+/GB
   SET COLOR OF BOX TO RG+/GB
   SET COLOR OF INFORMATION TO B/W
   SET COLOR OF FIELDS TO N/GB
ENDIF
CLEAR

*-- Sign-on banner

SET BORDER TO
@ 5,9 TO 16,69 DOUBLE COLOR RG+/GB
@ 8,10 SAY "            SHIPBOARD SECURITY FORCE SYSTEM"
@ 10,10 SAY "                    NWMS SUBMODULE"
@ 6,10 FILL TO 15,68 COLOR W+/N
@ 24,30 SAY " Press "+CHR(17)+CHR(196)+CHR(217)+" to
continue. "
gn_ikey=INKEY(500)

CLEAR

*-- Prevents clearing of menus from commands:
```

```
*-- SET STATUS and SET SCOREBOARD
DEFINE WINDOW FullScr FROM 0,0 TO 24,79 NONE
*-- Position at runtime and batch process
DEFINE WINDOW Savescr FROM 0,0 TO 21,79 NONE
*-- F1 Help
DEFINE WINDOW Helpscr FROM 0,0 TO 21,79 NONE
IF gn_ApGen=1
   *-- Are you sure? (exit application)
   DEFINE WINDOW Exit_App FROM 11,17 TO 15,62 DOUBLE
   *-- Pause message box
   DEFINE WINDOW Pause FROM 15,00 TO 19,79 DOUBLE
ENDIF

ACTIVATE WINDOW FullScr
@ 24,00
@ 23,00 SAY "Loading..."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU MEMMAIN MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF MEMMAIN PROMPT "SF REQS" AT 1,1
ON PAD PAD_1 OF MEMMAIN ACTIVATE POPUP REQSSF
DEFINE PAD PAD_2 OF MEMMAIN PROMPT "SF ROSTER" AT 1,11
ON PAD PAD_2 OF MEMMAIN ACTIVATE POPUP ROSTER
DEFINE PAD PAD_3 OF MEMMAIN PROMPT "SF DRILLS" AT 1,23
ON PAD PAD_3 OF MEMMAIN ACTIVATE POPUP DRILLS
DEFINE PAD PAD_4 OF MEMMAIN PROMPT "PRINT SF" AT 1,35
ON PAD PAD_4 OF MEMMAIN ACTIVATE POPUP PRINTSF
DEFINE PAD PAD_5 OF MEMMAIN PROMPT "QUALIFICATIONS" AT 1,46
ON PAD PAD_5 OF MEMMAIN ACTIVATE POPUP QUALS
DEFINE PAD PAD_6 OF MEMMAIN PROMPT "UTILITIES" AT 1,63
ON PAD PAD_6 OF MEMMAIN ACTIVATE POPUP UTILS
DEFINE PAD PAD_7 OF MEMMAIN PROMPT "EXIT " AT 1,74
ON PAD PAD_7 OF MEMMAIN ACTIVATE POPUP EXIT
?? "."
SET BORDER TO DOUBLE
*-- Popup
DEFINE POPUP REQSSF FROM 4,0 TO 8,29 ;
MESSAGE 'Position: '+CHR(27)+CHR(26)+CHR(25)+CHR(24)+'
Select: '+CHR(17)+CHR(196)+CHR(217)+'  Help: F1'
DEFINE BAR 1 OF REQSSF PROMPT "VIEW SPECIFIC REQUIREMENT ->"

DEFINE BAR 2 OF REQSSF PROMPT "EDIT REQUIREMENTS         ->"

DEFINE BAR 3 OF REQSSF PROMPT "PRINT REQUIREMENTS        ->"

ON SELECTION POPUP REQSSF DO ACT02
?? "."
SET BORDER TO DOUBLE
*-- Popup
DEFINE POPUP ROSTER FROM 4,11 TO 8,36 ;
```

```
MESSAGE 'Position: '+CHR(27)+CHR(26)+CHR(25)+CHR(24)+'
Select: '+CHR(17)+CHR(196)+CHR(217)+'  Help: F1'
DEFINE BAR 1 OF ROSTER PROMPT "VIEW SECURITY ROSTER"
DEFINE BAR 2 OF ROSTER PROMPT "UPDATE SECURITY ROSTER"
DEFINE BAR 3 OF ROSTER PROMPT "PRINT SECURITY ROSTER"
ON SELECTION POPUP ROSTER DO ACT03
?? "."
SET BORDER TO DOUBLE
*-- Popup
DEFINE POPUP DRILLS FROM 4,23 TO 8,37 ;
MESSAGE 'Position: '+CHR(27)+CHR(26)+CHR(25)+CHR(24)+'
Select: '+CHR(17)+CHR(196)+CHR(217)+'  Help: F1'
DEFINE BAR 1 OF DRILLS PROMPT "VIEW DRILLS"
DEFINE BAR 2 OF DRILLS PROMPT "EDIT DRILLS"
DEFINE BAR 3 OF DRILLS PROMPT "PRINT DRILLS"
ON SELECTION POPUP DRILLS DO ACT04
?? "."
SET BORDER TO DOUBLE
*-- Popup
DEFINE POPUP PRINTSF FROM 4,35 TO 10,58 ;
MESSAGE 'Position: '+CHR(27)+CHR(26)+CHR(25)+CHR(24)+'
Select: '+CHR(17)+CHR(196)+CHR(217)+'  Help: F1'
DEFINE BAR 1 OF PRINTSF PROMPT "PRINT REQUIREMENTS  ->"
DEFINE BAR 2 OF PRINTSF PROMPT "PRINT ROSTER"
DEFINE BAR 3 OF PRINTSF PROMPT "PRINT DRILL REPORT"
DEFINE BAR 4 OF PRINTSF PROMPT "PRINT XQUALS <=45   ->"
DEFINE BAR 5 OF PRINTSF PROMPT "PRINT EXPIRED QUALS ->"
ON SELECTION POPUP PRINTSF DO ACT05
?? "."
SET BORDER TO DOUBLE
*-- Popup
DEFINE POPUP QUALS FROM 4,46 TO 7,73 ;
MESSAGE 'Position: '+CHR(27)+CHR(26)+CHR(25)+CHR(24)+'
Select: '+CHR(17)+CHR(196)+CHR(217)+'  Help: F1'
DEFINE BAR 1 OF QUALS PROMPT "QUALS 45 DAYS TO EXPIRE ->"
DEFINE BAR 2 OF QUALS PROMPT "QUALS EXPIRED           ->"
ON SELECTION POPUP QUALS DO ACT06
?? "."
SET BORDER TO DOUBLE
*-- Popup
DEFINE POPUP UTILS FROM 4,58 TO 9,79 ;
MESSAGE 'Position: '+CHR(27)+CHR(26)+CHR(25)+CHR(24)+'
Select: '+CHR(17)+CHR(196)+CHR(217)+'  Help: F1'
DEFINE BAR 1 OF UTILS PROMPT "CHANGE SYSTEM DATE"
DEFINE BAR 2 OF UTILS PROMPT "BACKUP TO DRIVE A"
DEFINE BAR 3 OF UTILS PROMPT "COPY SF APPL DRIVE A"
DEFINE BAR 4 OF UTILS PROMPT "REPAIR INDEXES"
ON SELECTION POPUP UTILS DO ACT07
?? "."
SET BORDER TO DOUBLE
*-- Popup
```

```
DEFINE POPUP EXIT FROM 4,66 TO 7,79 ;
MESSAGE 'Position: '+CHR(27)+CHR(26)+CHR(25)+CHR(24)+'
Select: '+CHR(17)+CHR(196)+CHR(217)+'  Help: F1'
DEFINE BAR 1 OF EXIT PROMPT "TO dBASE IV"
DEFINE BAR 2 OF EXIT PROMPT "EXIT TO DOS"
ON SELECTION POPUP EXIT DO ACT08
?? "."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU REQSALL MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF REQSALL PROMPT "SAT REQS" AT 7,1
ON SELECTION PAD PAD_1 OF REQSALL DO ACT09
DEFINE PAD PAD_2 OF REQSALL PROMPT "BAF REQS" AT 7,12
ON SELECTION PAD PAD_2 OF REQSALL DO ACT09
DEFINE PAD PAD_3 OF REQSALL PROMPT "RESFOR REQS" AT 7,23
ON SELECTION PAD PAD_3 OF REQSALL DO ACT09
DEFINE PAD PAD_4 OF REQSALL PROMPT "AUGFOR REQS" AT 7,37
ON SELECTION PAD PAD_4 OF REQSALL DO ACT09
DEFINE PAD PAD_5 OF REQSALL PROMPT "DRILL REQS" AT 7,51
ON SELECTION PAD PAD_5 OF REQSALL DO ACT09
DEFINE PAD PAD_6 OF REQSALL PROMPT "EXIT" AT 7,64
ON SELECTION PAD PAD_6 OF REQSALL DO ACT09
?? "."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU EDREQS MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF EDREQS PROMPT "SAT REQS" AT 8,1
ON SELECTION PAD PAD_1 OF EDREQS DO ACT010
DEFINE PAD PAD_2 OF EDREQS PROMPT "BAF REQS" AT 8,12
ON SELECTION PAD PAD_2 OF EDREQS DO ACT010
DEFINE PAD PAD_3 OF EDREQS PROMPT "RESFOR REQS" AT 8,23
ON SELECTION PAD PAD_3 OF EDREQS DO ACT010
DEFINE PAD PAD_4 OF EDREQS PROMPT "AUGFOR REQS" AT 8,37
ON SELECTION PAD PAD_4 OF EDREQS DO ACT010
DEFINE PAD PAD_5 OF EDREQS PROMPT "DRILL REQS" AT 8,51
ON SELECTION PAD PAD_5 OF EDREQS DO ACT010
DEFINE PAD PAD_6 OF EDREQS PROMPT "EXIT" AT 8,65
ON SELECTION PAD PAD_6 OF EDREQS DO ACT010
?? "."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU PRNTREQS MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF PRNTREQS PROMPT "SAT REQS" AT 9,1
ON SELECTION PAD PAD_1 OF PRNTREQS DO ACT011
DEFINE PAD PAD_2 OF PRNTREQS PROMPT "BAF REQS" AT 9,12
ON SELECTION PAD PAD_2 OF PRNTREQS DO ACT011
DEFINE PAD PAD_3 OF PRNTREQS PROMPT "RESFOR REQS" AT 9,23
ON SELECTION PAD PAD_3 OF PRNTREQS DO ACT011
```

```
DEFINE PAD PAD_4 OF PRNTREQS PROMPT "AUGFOR REQS" AT 9,37
ON SELECTION PAD PAD_4 OF PRNTREQS DO ACT011
DEFINE PAD PAD_5 OF PRNTREQS PROMPT "DRILL REQS" AT 9,51
ON SELECTION PAD PAD_5 OF PRNTREQS DO ACT011
?? "."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU PEXPIRE MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF PEXPIRE PROMPT "SHOTGUN" AT 10,36
ON SELECTION PAD PAD_1 OF PEXPIRE DO ACT012
DEFINE PAD PAD_2 OF PEXPIRE PROMPT "45CAL" AT 10,46
ON SELECTION PAD PAD_2 OF PEXPIRE DO ACT012
DEFINE PAD PAD_3 OF PEXPIRE PROMPT "9MM" AT 10,54
ON SELECTION PAD PAD_3 OF PEXPIRE DO ACT012
DEFINE PAD PAD_4 OF PEXPIRE PROMPT "M14/16" AT 10,60
ON SELECTION PAD PAD_4 OF PEXPIRE DO ACT012
?? "."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU PEXPIRED MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF PEXPIRED PROMPT "SHOTGUN" AT 11,36
ON SELECTION PAD PAD_1 OF PEXPIRED DO ACT013
DEFINE PAD PAD_2 OF PEXPIRED PROMPT "45CAL" AT 11,46
ON SELECTION PAD PAD_2 OF PEXPIRED DO ACT013
DEFINE PAD PAD_3 OF PEXPIRED PROMPT "9MM" AT 11,54
ON SELECTION PAD PAD_3 OF PEXPIRED DO ACT013
DEFINE PAD PAD_4 OF PEXPIRED PROMPT "M14/16" AT 11,60
ON SELECTION PAD PAD_4 OF PEXPIRED DO ACT013
?? "."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU EXPIRE MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF EXPIRE PROMPT "SHOTGUN" AT 7,47
ON SELECTION PAD PAD_1 OF EXPIRE DO ACT014
DEFINE PAD PAD_2 OF EXPIRE PROMPT "45CAL" AT 7,57
ON SELECTION PAD PAD_2 OF EXPIRE DO ACT014
DEFINE PAD PAD_3 OF EXPIRE PROMPT "9MM" AT 7,65
ON SELECTION PAD PAD_3 OF EXPIRE DO ACT014
DEFINE PAD PAD_4 OF EXPIRE PROMPT "M14/M16" AT 7,71
ON SELECTION PAD PAD_4 OF EXPIRE DO ACT014
?? "."
SET BORDER TO DOUBLE
*-- Bar
DEFINE MENU EXPIRED MESSAGE 'Position with:
'+CHR(27)+CHR(26)+' - <Enter> to select choice - <F1> Help'
DEFINE PAD PAD_1 OF EXPIRED PROMPT "SHOTGUN" AT 8,47
ON SELECTION PAD PAD_1 OF EXPIRED DO ACT015
DEFINE PAD PAD_2 OF EXPIRED PROMPT "45 CAL" AT 8,57
```

```
ON SELECTION PAD PAD_2 OF EXPIRED DO ACT015
DEFINE PAD PAD_3 OF EXPIRED PROMPT "9MM" AT 8,66
ON SELECTION PAD PAD_3 OF EXPIRED DO ACT015
DEFINE PAD PAD_4 OF EXPIRED PROMPT "M14/16" AT 8,72
ON SELECTION PAD PAD_4 OF EXPIRED DO ACT015
?? "."
RETURN
*-- EOP: MPDEF.PRG

PROCEDURE 1HELP1
ln_key=INKEY()
ON KEY LABEL F1
lc_popmenu=IIF( "" = POPUP(), MENU(), POPUP() )
ACTIVATE WINDOW Helpscr
SET ESCAPE OFF
ACTIVATE SCREEN
@ 0,0 CLEAR TO 21,79
@ 1,0 TO 21,79 COLOR RG+/GB
@ 24,00
@ 24,26 SAY "Press any key to continue..."
@ 0,0 SAY ""
DO CASE
*-- help for menu MEMMAIN
CASE "MEMMAIN" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu REQSSF
CASE "REQSSF" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu ROSTER
CASE "ROSTER" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu DRILLS
CASE "DRILLS" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu PRINTSF
CASE "PRINTSF" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu QUALS
CASE "QUALS" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu UTILS
CASE "UTILS" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu EXIT
CASE "EXIT" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu REQSALL
CASE "REQSALL" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu EDREQS
CASE "EDREQS" = lc_popmenu
```

```
   @ 2,2 SAY "No Help defined."
*-- help for menu PRNTREQS
CASE "PRNTREQS" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu PEXPIRE
CASE "PEXPIRE" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu PEXPIRED
CASE "PEXPIRED" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu EXPIRE
CASE "EXPIRE" = lc_popmenu
   @ 2,2 SAY "No Help defined."
*-- help for menu EXPIRED
CASE "EXPIRED" = lc_popmenu
   @ 2,2 SAY "No Help defined."
OTHERWISE
   @ 2,2 SAY "Unknown menu name, help was never defined."
ENDCASE
ln_key=INKEY(0)
SET ESCAPE ON
@ 24,00
DEACTIVATE WINDOW Helpscr
ON KEY LABEL F1 DO lHELP1
RETURN
*-- EOP: lHELP1


************************************************************
**********
* Program......: MEMMAIN.PRG
* Author.......: Sidney R. Settlemyer, Lt. USN
* Date.........: 7-23-91
* Notice.......: Written as partial fulfillment of thesis
* dBASE Ver....:
* Generated by.: APGEN version 1.3
* Description..: BAR MENU FOR THE SECURITY FORCE PROGRAM

* Description..: Menu actions
************************************************************
**********
PROCEDURE MEMMAIN
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="01"
SET COLOR OF NORMAL TO W+/B
CLEAR
PRIVATE lc_ApGen
lc_ApGen=LTRIM(STR(gn_ApGen))

DO SET01
```

```
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE MENU MEMMAIN

@ 0,0 CLEAR TO 2,79

*-- After menu

RETURN
*-- EOP MEMMAIN

PROCEDURE SET01
ON KEY LABEL F1 DO 1HELP1

DO DBF01 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 0,0 TO 2,79 DOUBLE COLOR RG+/GB
   @ 1,1 CLEAR TO 1,78
   @ 1,1 FILL TO 1,78 COLOR W+/N
   @ 1,1 SAY "SF REQS" COLOR W+/N
   @ 1,11 SAY "SF ROSTER" COLOR W+/N
   @ 1,23 SAY "SF DRILLS" COLOR W+/N
   @ 1,35 SAY "PRINT SF" COLOR W+/N
   @ 1,46 SAY "QUALIFICATIONS" COLOR W+/N
   @ 1,63 SAY "UTILITIES" COLOR W+/N
   @ 1,74 SAY "EXIT " COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF01
CLOSE DATABASES
```

```
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT01
*-- Begin MEMMAIN: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
   lc_new='Y'
   DO REQSSF WITH " 01"
CASE "PAD_2" = PAD()
   lc_new='Y'
   DO ROSTER WITH " 01"
CASE "PAD_3" = PAD()
   lc_new='Y'
   DO DRILLS WITH " 01"
CASE "PAD_4" = PAD()
   lc_new='Y'
   DO PRINTSF WITH " 01"
CASE "PAD_5" = PAD()
   lc_new='Y'
   DO QUALS WITH " 01"
CASE "PAD_6" = PAD()
   lc_new='Y'
   DO UTILS WITH " 01"
CASE "PAD_7" = PAD()
   lc_new='Y'
   DO EXIT WITH " 01"
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && MEMMAIN
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
*****************************************************************
**********
* Program......: REQSSF.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: PROVIDES USER WITH THE SECURITY
REQUIREMENTS FOR HIS SHIP

* Description..: Menu actions
*****************************************************************
**********
PROCEDURE REQSSF
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="02"

DO SET02
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE POPUP REQSSF

*-- After menu

RETURN
*-- EOP REQSSF
```

```
PROCEDURE SET02
ON KEY LABEL F1 DO 1HELP1

DO DBF02 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF
ENDIF
RETURN

PROCEDURE DBF02
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT02
*-- Begin REQSSF: POPUP Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE BAR() = 1
```

```
   lc_new='Y'
   DO REQSALL WITH "B02"
CASE BAR() = 2
   lc_new='Y'
   DO EDREQS WITH "B02"
CASE BAR() = 3
   lc_new='Y'
   DO PRNTREQS WITH "B02"
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE POPUP && REQSSF
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
************************************************************
**********
* Program......: ROSTER.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment for thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: WILL PROVIDE OPTIONS ON ROSTER INFO FOR
SECURITY FORCE

* Description..: Menu actions
************************************************************
**********
PROCEDURE ROSTER
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="03"

DO SET03
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE POPUP ROSTER
```

```
*-- After menu

RETURN
*-- EOP ROSTER

PROCEDURE SET03
ON KEY LABEL F1 DO 1HELP1

DO DBF03 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF
ENDIF
RETURN

PROCEDURE DBF03
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT03
*-- Begin ROSTER: POPUP Menu Actions.
*-- (before item, action, and after item)
*
```

```
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE BAR() = 1
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE NOAPPEND NODELETE NOEDIT FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 2
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   *-- Desc: attach format file MEM_INFO
   SET FORMAT TO MEM_INFO
   APPEND

   *-- close format file so as not to affect READ's
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 3
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET PRINT ON
   REPORT FORM SFROSTER.FRM PLAIN
   SET PRINT OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE POPUP && ROSTER
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
************************************************************
**********
* Program......: DRILLS.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Writtrn in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: ALLOWS USER TO ACCESS DRILLS

* Description..: Menu actions
************************************************************
**********
PROCEDURE DRILLS
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="04"

DO SET04
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE POPUP DRILLS

*-- After menu

RETURN
*-- EOP DRILLS

PROCEDURE SET04
ON KEY LABEL F1 DO 1HELP1

DO DBF04 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
```

```
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF
ENDIF
RETURN

PROCEDURE DBF04
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT04
*-- Begin DRILLS: POPUP Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE BAR() = 1
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE DRILLS
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening DRILLS.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO drills
   *-- Desc: Browse file - DRILLS
   BROWSE NOAPPEND NODELETE NOEDIT FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 2
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE DRILLS
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening DRILLS.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   *-- Desc: attach format file DRILLS
   SET FORMAT TO DRILLS
   APPEND
   *-- close format file so as not to affect READ's
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 3
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE DRILLS
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening DRILLS.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET PRINT ON
   REPORT FORM DRILREP PLAIN
   SET PRINT OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE POPUP && DRILLS
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
IF lc_dbf='Y' .AND. .NOT. lc_new='Y'
   lc_file="DBF"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
************************************************************
**********
* Program......: PRINTSF.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: THIS WILL PROVIDE PRINT OPTIONS FOR ROSTER
AND REQUIREMENTS

* Description..: Menu actions
************************************************************
**********
PROCEDURE PRINTSF
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="05"

DO SET05
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE POPUP PRINTSF

*-- After menu

RETURN
*-- EOP PRINTSF

PROCEDURE SET05
ON KEY LABEL F1 DO 1HELP1

DO DBF05 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF
ENDIF
RETURN

PROCEDURE DBF05
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT05
*-- Begin PRINTSF: POPUP Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE BAR() = 1
   lc_new='Y'
   DO PRNTREQS WITH "B05"
CASE BAR() = 2
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET PRINT ON
   REPORT FORM SFROSTER.FRM PLAIN
   SET PRINT OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 3
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE DRILLS
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening DRILLS.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET PRINT ON
   REPORT FORM DRILREP PLAIN
   SET PRINT OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 4
   lc_new='Y'
   DO PEXPIRE WITH "B05"
CASE BAR() = 5
   lc_new='Y'
   DO PEXPIRED WITH "B05"
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE POPUP && PRINTSF
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
IF lc_dbf='Y' .AND. .NOT. lc_new='Y'
```

```
   lc_file="DBF"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
************************************************************
**********
* Program......: QUALS.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: PROVIDES USER WITH OPTIONS ON VIEWING SMALL
ARMS QUALS EXPIRING

* Description..: Menu actions
************************************************************
**********
PROCEDURE QUALS
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="06"

DO SET06
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE POPUP QUALS

*-- After menu

RETURN
*-- EOP QUALS

PROCEDURE SET06
ON KEY LABEL F1 DO 1HELP1

DO DBF06 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF
ENDIF
RETURN

PROCEDURE DBF06
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT06
*-- Begin QUALS: POPUP Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE BAR() = 1
   lc_new='Y'
   DO EXPIRE WITH "B06"
CASE BAR() = 2
   lc_new='Y'
   DO EXPIRED WITH "B06"
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE POPUP && QUALS
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
***************************************************************
**********
* Program......: UTILS.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: ALLOWS USER TO SELECT UTILITIES FROM MAIN
MENU

* Description..: Menu actions
***************************************************************
**********
PROCEDURE UTILS
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="07"

DO SET07
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE POPUP UTILS

*-- After menu

RETURN
*-- EOP UTILS

PROCEDURE SET07
ON KEY LABEL F1 DO 1HELP1

DO DBF07 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF
ENDIF
RETURN

PROCEDURE DBF07
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT07
*-- Begin UTILS: POPUP Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE BAR() = 1
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   RUN DATE

   IF .NOT. gl_batch
      DO AftAct
   ENDIF
```

```
CASE BAR() = 2
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   RUN COPY MEM_INFO.*

   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 3
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   RUN COPY *.* A:

   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE BAR() = 4
   SET EXCLUSIVE ON
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE MEM_INFO
   IF "" <> DBF()
      SET INDEX TO MEM_INFO
   ENDIF
   SET ORDER TO LASTNAME
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   SET EXCLUSIVE OFF
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   lc_say='Reindexing Database...'
   DO info_box WITH lc_say
   SET TALK ON
   REINDEX
   SET TALK OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE POPUP && UTILS
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
*****************************************************************
**********
* Program......: EXIT.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: WILL ALLOW USER TO EXIT SECURITY SYSTEM

* Description..: Menu actions
*****************************************************************
**********
PROCEDURE EXIT
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="08"

DO SET08
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE POPUP EXIT

*-- After menu

RETURN
*-- EOP EXIT
```

```
PROCEDURE SET08
ON KEY LABEL F1 DO 1HELP1

DO DBF08 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF
ENDIF
RETURN

PROCEDURE DBF08
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT08
*-- Begin EXIT: POPUP Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE BAR() = 1
```

```
   *-- Return to caller
   gc_quit='Q'
   IF LEFT(entryflg,1) <> "B"
      DEACTIVATE POPUP && EXIT
   ELSE
      DEACTIVATE MENU
   ENDIF
   RETURN
CASE BAR() = 2
   *-- Quit dBASE
   CLOSE DATABASES
   QUIT
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE POPUP && EXIT
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
************************************************************
**********
* Program......: REQSALL.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: DISPLAYS OPTIONS FOR SF REQUIREMENTS

* Description..: Menu actions
************************************************************
**********
PROCEDURE REQSALL
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="09"

DO SET09
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code
```

```
ACTIVATE MENU REQSALL

@ 6,0 CLEAR TO 8,79

*-- After menu

RETURN
*-- EOP REQSALL

PROCEDURE SET09
ON KEY LABEL F1 DO 1HELP1

DO DBF09 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 6,0 TO 8,79 DOUBLE COLOR RG+/GB
   @ 7,1 CLEAR TO 7,78
   @ 7,1 FILL TO 7,78 COLOR W+/N
   @ 7,1 SAY "SAT REQS" COLOR W+/N
   @ 7,12 SAY "BAF REQS" COLOR W+/N
   @ 7,23 SAY "RESFOR REQS" COLOR W+/N
   @ 7,37 SAY "AUGFOR REQS" COLOR W+/N
   @ 7,51 SAY "DRILL REQS" COLOR W+/N
   @ 7,64 SAY "EXIT" COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF09
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE REQUIREM
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening REQUIREM.DBF"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT09
*-- Begin REQSALL: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE REQUIREM
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening REQUIREM.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO satreqs
   *-- Desc: Browse file - REQUIREM
   BROWSE NOMENU NOAPPEND NODELETE NOEDIT FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_2" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO bafreqs
   *-- Desc: Browse file -
   BROWSE NOMENU NOAPPEND NODELETE NOEDIT FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_3" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO resreqs
   *-- Desc: Browse file -
   BROWSE NOMENU NOAPPEND NODELETE NOEDIT FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_4" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO augreqs
   *-- Desc: Browse file -
   BROWSE NOMENU NOAPPEND NODELETE NOEDIT FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_5" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO drlreqs
   *-- Desc: Browse file -
   BROWSE FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
```

```
      DO AftAct
   ENDIF
CASE "PAD_6" = PAD()
   *-- Return to caller
   gc_quit='Q'
   DEACTIVATE MENU && REQSALL
   RETURN
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && REQSALL
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
************************************************************
**********
* Program......: EDREQS.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment for thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: ALLOWS USER TO EDIT SSF REQUIREMENTS

* Description..: Menu actions
************************************************************
**********
PROCEDURE EDREQS
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="010"

DO SET010
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code
```

```
ACTIVATE MENU EDREQS

@ 7,0 CLEAR TO 9,79

*-- After menu

RETURN
*-- EOP EDREQS

PROCEDURE SET010
ON KEY LABEL F1 DO 1HELP1

DO DBF010 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 7,0 TO 9,79 DOUBLE COLOR RG+/GB
   @ 8,1 CLEAR TO 8,78
   @ 8,1 FILL TO 8.78 COLOR W+/N
   @ 8,1 SAY "SAT REQS" COLOR W+/N
   @ 8,12 SAY "BAF REQS" COLOR W+/N
   @ 8,23 SAY "RESFOR REQS" COLOR W+/N
   @ 8,37 SAY "AUGFOR REQS" COLOR W+/N
   @ 8,51 SAY "DRILL REQS" COLOR W+/N
   @ 8,65 SAY "EXIT" COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF010
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
```

```
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT010
*-- Begin EDREQS: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE REQUIREM
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening REQUIREM.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO satreqs
   *-- Desc: Browse file - REQUIREM
   BROWSE FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_2" = PAD()
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE REQUIREM
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening REQUIREM.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO bafreqs
   *-- Desc: Browse file - REQUIREM
   BROWSE FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_3" = PAD()
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE REQUIREM
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening REQUIREM.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
```

```
      RELEASE lc_message
      IF .NOT. gl_batch
         DO BefAct
      ENDIF
      SET SCOREBOARD ON
      SET MESSAGE TO
      SET FORMAT TO resreqs
      *-- Desc: Browse file - REQUIREM
      BROWSE FORMAT
      SET FORMAT TO
      SET SCOREBOARD OFF
      IF .NOT. gl_batch
         DO AftAct
      ENDIF
   CASE "PAD_4" = PAD()
      *-- Open Item level view/database and indexes
      CLOSE DATABASES
      lc_dbf='Y'
      lc_message="0"
      ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
      USE REQUIREM
      ON ERROR
      gn_error=VAL(lc_message)
      IF gn_error > 0
         DO Pause WITH ;
         "Error opening REQUIREM.DBF"
         gn_error=0
         lc_file="SET"+gc_prognum
         DO &lc_file.
         RETURN
      ENDIF
      lc_new='Y'
      RELEASE lc_message
      IF .NOT. gl_batch
         DO BefAct
      ENDIF
      SET SCOREBOARD ON
      SET MESSAGE TO
      SET FORMAT TO augreqs
      *-- Desc: Browse file - REQUIREM
      BROWSE FORMAT
      SET FORMAT TO
      SET SCOREBOARD OFF
      IF .NOT. gl_batch
         DO AftAct
      ENDIF
   CASE "PAD_5" = PAD()
      *-- Open Item level view/database and indexes
      CLOSE DATABASES
      lc_dbf='Y'
      lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE REQUIREM
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening REQUIREM.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FORMAT TO drlreqs
   *-- Desc: Browse file - REQUIREM
   BROWSE FORMAT
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_6" = PAD()
   *-- Return to caller
   gc_quit='Q'
   DEACTIVATE MENU && EDREQS
   RETURN
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && EDREQS
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
IF lc_dbf='Y' .AND. .NOT. lc_new='Y'
```

```
   lc_file="DBF"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
*****************************************************************
**********
* Program......: PRNTREQS.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment for thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: WILL ALLOW USER TO PRINT SPECIFIC SSF
RQUIREMENTS

* Description..: Menu actions
*****************************************************************
**********
PROCEDURE PRNTREQS
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="011"

DO SET011
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE MENU PRNTREQS

@ 8,0 CLEAR TO 10,62

*-- After menu

RETURN
*-- EOP PRNTREQS

PROCEDURE SET011
ON KEY LABEL F1 DO 1HELP1

DO DBF011 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 8,0 TO 10,62 DOUBLE COLOR RG+/GB
   @ 9,1 CLEAR TO 9,61
   @ 9,1 FILL TO 9,61 COLOR W+/N
   @ 9,1 SAY "SAT REQS" COLOR W+/N
   @ 9,12 SAY "BAF REQS" COLOR W+/N
   @ 9,23 SAY "RESFOR REQS" COLOR W+/N
   @ 9,37 SAY "AUGFOR REQS" COLOR W+/N
   @ 9,51 SAY "DRILL REQS" COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF011
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT011
*-- Begin PRNTREQS: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
```

```
      *-- Open Item level view/database and indexes
      CLOSE DATABASES
      lc_dbf='Y'
      lc_message="0"
      ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
      USE REQUIREM
      ON ERROR
      gn_error=VAL(lc_message)
      IF gn_error > 0
         DO Pause WITH ;
         "Error opening REQUIREM.DBF"
         gn_error=0
         lc_file="SET"+gc_prognum
         DO &lc_file.
         RETURN
      ENDIF
      lc_new='Y'
      RELEASE lc_message
      IF .NOT. gl_batch
         DO BefAct
      ENDIF
      SET MESSAGE TO
      *-- Desc: List [<parameters>]
      CLEAR
      SET PRINT ON
      DISPLAY ALL FIELDS sat
      SET PRINT OFF
      IF .NOT. gl_batch
         DO AftAct
      ENDIF
   CASE "PAD_2" = PAD()
      *-- Open Item level view/database and indexes
      CLOSE DATABASES
      lc_dbf='Y'
      lc_message="0"
      ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
      USE REQUIREM
      ON ERROR
      gn_error=VAL(lc_message)
      IF gn_error > 0
         DO Pause WITH ;
         "Error opening REQUIREM.DBF"
         gn_error=0
         lc_file="SET"+gc_prognum
         DO &lc_file.
         RETURN
      ENDIF
      lc_new='Y'
      RELEASE lc_message
      IF .NOT. gl_batch
         DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: List [<parameters>]
   CLEAR
   SET PRINT ON
   DISPLAY ALL FIELDS baf
   SET PRINT OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_3" = PAD()
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE REQUIREM
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
      DO Pause WITH ;
      "Error opening REQUIREM.DBF"
      gn_error=0
      lc_file="SET"+gc_prognum
      DO &lc_file.
      RETURN
   ENDIF
   lc_new='Y'
   RELEASE lc_message
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: List [<parameters>]
   CLEAR
   SET PRINT ON
   DISPLAY ALL FIELDS resfor
   SET PRINT OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_4" = PAD()
   *-- Open Item level view/database and indexes
   CLOSE DATABASES
   lc_dbf='Y'
   lc_message="0"
   ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
   USE REQUIREM
   ON ERROR
   gn_error=VAL(lc_message)
   IF gn_error > 0
```

```
         DO Pause WITH ;
         "Error opening REQUIREM.DBF"
         gn_error=0
         lc_file="SET"+gc_prognum
         DO &lc_file.
         RETURN
      ENDIF
      lc_new='Y'
      RELEASE lc_message
      IF .NOT. gl_batch
         DO BefAct
      ENDIF
      SET MESSAGE TO
      *-- Desc: List [<parameters>]
      CLEAR
      SET PRINT ON
      DISPLAY ALL FIELDS augfor
      SET PRINT OFF
      IF .NOT. gl_batch
         DO AftAct
      ENDIF
   CASE "PAD_5" = PAD()
      *-- Open Item level view/database and indexes
      CLOSE DATABASES
      lc_dbf='Y'
      lc_message="0"
      ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
      USE REQUIREM
      ON ERROR
      gn_error=VAL(lc_message)
      IF gn_error > 0
         DO Pause WITH ;
         "Error opening REQUIREM.DBF"
         gn_error=0
         lc_file="SET"+gc_prognum
         DO &lc_file.
         RETURN
      ENDIF
      lc_new='Y'
      RELEASE lc_message
      IF .NOT. gl_batch
         DO BefAct
      ENDIF
      SET MESSAGE TO
      *-- Desc: List [<parameters>]
      CLEAR
      SET PRINT ON
      DISPLAY ALL FIELDS sf_drills
      SET PRINT OFF
      IF .NOT. gl_batch
         DO AftAct
   ENDIF
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && PRNTREQS
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
IF lc_dbf='Y' .AND. .NOT. lc_new='Y'
   lc_file="DBF"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
***************************************************************
**********
* Program......: PEXPIRE.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..:

* Description..: Menu actions
***************************************************************
**********
PROCEDURE PEXPIRE
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="012"

DO SET012
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code
```

```
ACTIVATE MENU PEXPIRE

@ 9,35 CLEAR TO 11,66

*-- After menu

RETURN
*-- EOP PEXPIRE

PROCEDURE SET012
ON KEY LABEL F1 DO 1HELP1

DO DBF012 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 9,35 TO 11,66 DOUBLE COLOR RG+/GB
   @ 10,36 CLEAR TO 10,65
   @ 10,36 FILL TO 10,65 COLOR W+/N
   @ 10,36 SAY "SHOTGUN" COLOR W+/N
   @ 10,46 SAY "45CAL" COLOR W+/N
   @ 10,54 SAY "9MM" COLOR W+/N
   @ 10,60 SAY "M14/16" COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF012
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
```

```
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT012
*-- Begin PEXPIRE: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO SHTGN_DATE<=DATE()-320
   GOTO TOP
   SET PRINT ON
   REPORT FORM SPEXPIRE PLAIN
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_2" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO F45_DATE<=DATE()-320
   GOTO TOP
   SET PRINT ON
   REPORT FORM FPEXPIRE PLAIN  NOEJECT
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_3" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO F9MM_DATE<=DATE()-320
   GOTO TOP
   SET PRINT ON
   REPORT FORM NPEXPIRE PLAIN  NOEJECT
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_4" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO M14_16DATE<=DATE()-320
   GOTO TOP
   SET PRINT ON
   REPORT FORM MPEXPIRE PLAIN  NOEJECT
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && PEXPIRE
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
************************************************************
**********
* Program......: PEXPIRED.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..:

* Description..: Menu actions
************************************************************
**********
PROCEDURE PEXPIRED
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="013"

DO SET013
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE MENU PEXPIRED

@ 10,35 CLEAR TO 12,66

*-- After menu

RETURN
*-- EOP PEXPIRED

PROCEDURE SET013
ON KEY LABEL F1 DO 1HELP1

DO DBF013 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 10,35 TO 12,66 DOUBLE COLOR RG+/GB
   @ 11,36 CLEAR TO 11,65
   @ 11,36 FILL TO 11,65 COLOR W+/N
   @ 11,36 SAY "SHOTGUN" COLOR W+/N
   @ 11,46 SAY "45CAL" COLOR W+/N
   @ 11,54 SAY "9MM" COLOR W+/N
   @ 11,60 SAY "M14/16" COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF013
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT013
*-- Begin PEXPIRED: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO SHTGN_DATE<=DATE()-365
   GOTO TOP
   SET PRINT ON
   REPORT FORM SPEXPIRE PLAIN  NOEJECT
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
```

```
      DO AftAct
   ENDIF
CASE "PAD_2" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO F45_DATE<=DATE()-365
   GOTO TOP
   SET PRINT ON
   REPORT FORM FPEXPIRE PLAIN  NOEJECT
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_3" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO F9MM_DATE<=DATE()-365
   GOTO TOP
   SET PRINT ON
   REPORT FORM NPEXPIRE PLAIN  NOEJECT
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_4" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET MESSAGE TO
   *-- Desc: Report
   SET FILTER TO M14_16DATE<=DATE()-365
   GOTO TOP
   SET PRINT ON
   REPORT FORM MPEXPIRE PLAIN  NOEJECT
   SET PRINT OFF
   SET FILTER TO
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && PEXPIRED
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
*************************************************************
**********
* Program......: EXPIRE.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: ALOOWS USER TO SELECT TO VIEW EXPIRING
QUALS FOR SMALL ARMS

* Description..: Menu actions
*************************************************************
**********
PROCEDURE EXPIRE
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="014"

DO SET014
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE MENU EXPIRE

@ 6,46 CLEAR TO 8,78

*-- After menu

RETURN
*-- EOP EXPIRE
```

```
PROCEDURE SET014
ON KEY LABEL F1 DO 1HELP1

DO DBF014 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 6,46 TO 8,78 DOUBLE COLOR RG+/GB
   @ 7,47 CLEAR TO 7,77
   @ 7,47 FILL TO 7,77 COLOR W+/N
   @ 7,47 SAY "SHOTGUN" COLOR W+/N
   @ 7,57 SAY "45CAL" COLOR W+/N
   @ 7,65 SAY "9MM" COLOR W+/N
   @ 7,71 SAY "M14/M16" COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF014
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN
```

```
PROCEDURE ACT014
*-- Begin EXPIRE: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO SHTGN_DATE<=DATE()-320
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,shtgn_qual,shtgn_date LOCK 2
FREEZE SHTGN_DATE NOAPPEND NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_2" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO F45_DATE<=DATE()-320
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,f45_qual,f45_date NOAPPEND
NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_3" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO F9MM_DATE<=DATE()-320
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,f9mm_qual,f9mm_date NOAPPEND
NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_4" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO M14_16DATE<=DATE()-320
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,m14_16qual,m14_16date
NOAPPEND NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && EXPIRE
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
```

```
RETURN
*****************************************************************
**********
* Program......: EXPIRED.PRG
* Author.......: Sidney R. Settlemyer
* Date.........: 7-23-91
* Notice.......: Written in partial fulfillment of thesis
* dBASE Ver....: 1.1
* Generated by.: APGEN version 1.3
* Description..: PROVIDE USER WITH OPTION TO VIEW PERSONS
WITH EXPIRED SA QUALS

* Description..: Menu actions
*****************************************************************
**********
PROCEDURE EXPIRED
PARAMETER entryflg
PRIVATE gc_prognum
gc_prognum="015"

DO SET015
IF gn_error > 0
   gn_error=0
   RETURN
ENDIF

*-- Before menu code


ACTIVATE MENU EXPIRED

@ 7,46 CLEAR TO 9,78

*-- After menu

RETURN
*-- EOP EXPIRED

PROCEDURE SET015
ON KEY LABEL F1 DO 1HELP1

DO DBF015 && open menu level database

IF gn_error = 0
   IF gl_color .AND. .NOT. SET("ATTRIBUTE") =
"W+/B,RG+/GB,N/N "+;
      CHR(38)+CHR(38)+" W+/N,W/B,RG+/GB,B/W,N/GB"
      SET COLOR OF NORMAL TO W+/B
      SET COLOR OF MESSAGES TO W+/N
      SET COLOR OF TITLES TO W/B
      SET COLOR OF HIGHLIGHT TO RG+/GB
      SET COLOR OF BOX TO RG+/GB
      SET COLOR OF INFORMATION TO B/W
      SET COLOR OF FIELDS TO N/GB
   ENDIF

   SET BORDER TO
   @ 7,46 TO 9,78 DOUBLE COLOR RG+/GB
   @ 8,47 CLEAR TO 8,77
   @ 8,47 FILL TO 8,77 COLOR W+/N
   @ 8,47 SAY "SHOTGUN" COLOR W+/N
   @ 8,57 SAY "45 CAL" COLOR W+/N
   @ 8,66 SAY "9MM" COLOR W+/N
   @ 8,72 SAY "M14/16" COLOR W+/N
ENDIF
RETURN

PROCEDURE DBF015
CLOSE DATABASES
*-- Open menu level view/database
lc_message="0"
ON ERROR lc_message=LTRIM(STR(ERROR()))+" "+MESSAGE()
USE MEM_INFO
IF "" <> DBF()
   SET INDEX TO MEM_INFO
ENDIF
SET ORDER TO LASTNAME
ON ERROR
gn_error=VAL(lc_message)
IF gn_error > 0
   DO Pause WITH ;
   "Error opening MEM_INFO.DBF or index(es) MEM_INFO"
   lc_new='Y'
   RETURN
ENDIF
lc_new='Y'
RELEASE lc_message
RETURN

PROCEDURE ACT015
*-- Begin EXPIRED: BAR Menu Actions.
*-- (before item, action, and after item)
*
PRIVATE lc_new, lc_dbf
lc_new=' '
lc_dbf=' '
DO CASE
CASE "PAD_1" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO SHTGN_DATE<=DATE()-365
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,shtgn_qual,shtgn_date
NOAPPEND NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_2" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO F45_DATE<=DATE()-365
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,f45_qual,f45_date LOCK 2
NOAPPEND NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_3" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO F9MM_DATE<=DATE()-365
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,f9mm_qual,f9mm_date LOCK 2
NOAPPEND NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
CASE "PAD_4" = PAD()
   IF .NOT. gl_batch
      DO BefAct
   ENDIF
   SET SCOREBOARD ON
   SET MESSAGE TO
   SET FILTER TO M14_16DATE<=DATE()-365
   GOTO TOP
   SET FORMAT TO mem_info
   *-- Desc: Browse file -
   BROWSE FIELDS
lastname,initials,rank_rate,ssn,m14_16qual,m14_16date LOCK 2
NOAPPEND NODELETE NOEDIT FORMAT
   SET FILTER TO
   SET FORMAT TO
   SET SCOREBOARD OFF
   IF .NOT. gl_batch
      DO AftAct
   ENDIF
OTHERWISE
   @ 24,00
   @ 24,21 SAY "This item has no action. Press a key."
   x=INKEY(0)
   @ 24,00
ENDCASE
SET MESSAGE TO
IF gc_quit='Q'
   IF LEFT(entryflg,1) = "B"
      DEACTIVATE MENU
   ELSE
      DEACTIVATE MENU && EXPIRED
   ENDIF
ENDIF
IF lc_new='Y'
   lc_file="SET"+gc_prognum
   DO &lc_file.
ENDIF
RETURN
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Superintendent Attn: Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Superintendent Attn: Professor T. X. Bui Code AS/Bd Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Commanding Officer NROTC Unit, RAS 104 Attn: CDR. B.B. Giannotti University of Texas Austin, TX 78712-1184	1
5.	Superintendent Naval Postgraduate School Code 37 Monterey, CA 93943-5000	1
6.	Director, Information Systems (OP-945) Office of the Chief of Naval Operations Navy Department Washington, DC 20350-2000	1
7.	Navy Personnel Research and Development Center Code 411 San Diego, CA 92152-6800	1
8.	Commander, Naval Surface Force, Pacific Fleet Attn: Code N64 San Diego, CA 92155-5035	1
9.	Nuclear Weapons Training Group Attn: LT Orr Code 20 NAS North Island San Diego, CA 92315	1

10. Nuclear Weapons Training Group 1
Technical Inspection Department
Naval Station Norfolk
Norfolk, VA 23511-6496

11. Commander, Naval Surface Forces 1
U.S. Atlantic Fleet
Attn: Code 515
Norfolk, VA 23511-6292

12. Naval Surface Warfare Center 1
White Oak
Attn: Code H101
10901 New Hampshire Avenue
Silver Spring, MD 20903-5000

13. Naval Ordnance Station, Louisville 1
Attn: Code 5051, N. Woods
Louisville, KY 40214-5001

~~[illegible]~~ 5
[illegible]

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

GAYLORD S

AMERICAN BINDERY
2
4-94

DUDLEY KNOX LIBRARY
3 2768 00018908 8